



**DEVELOPMENT OF 2-DIMENSIONAL CLOUD RISE MODEL
TO ANALYZE INITIAL NUCLEAR CLOUD RISE**

THESIS

Karson A. Sandman, Captain, USAF

AFIT/GNE/ENP/05-11

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

**DEVELOPMENT OF 2-DIMENSIONAL CLOUD RISE MODEL
TO ANALYZE INITIAL NUCLEAR CLOUD RISE**

THESIS

Presented to the Faculty
Department of Engineering Physics
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Nuclear Engineering

Karson A. Sandman, BS

Captain, USAF

March 2005

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

**DEVELOPMENT OF 2-DIMENSIONAL CLOUD RISE MODEL
TO ANALYZE INITIAL NUCLEAR CLOUD RISE**


Karson A. Sandman, B.S.
Capt, USAF

Approved:



Steven T. Fiorino (Chairman)

18 MAR 05
date



Charles J. Bridgman (Member)

3/11/05
date



Vincent J. Jodoin (Member)

11 Apr 05
date

Abstract

The objective of this research is to create a two-dimensional cloud rise model that could be used instead of the current one-dimensional cloud rise model in the Defense Land Fallout Interpretive Code (DELFI) option of the Hazard Prediction and Assessment Capability (HPAC). The model includes numerical analysis of partial differential equations involving pressure, potential temperature, horizontal and vertical winds, and specific humidity. The 2-D model developed provides a much more detailed definition of the physical properties within the mushroom cloud than the 1-D DELFI option. This is particularly useful in fallout studies on particle formation, fractionation, and particle location within the rising/risen cloud. The analysis model created for this study is the result of modifications to a convective cloud simulation. The primary modification to the convective cloud model is the incorporation of initial conditions for a nuclear cloud similar to those used in DELFI's initial conditions module. The code is compared to atmospheric test data for verification purposes.

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Lt Col Steven Fiorino, for his guidance and support throughout the course of this thesis effort. I also thank my committee members, Dr. Charles Bridgman and Lt Col Vincent Jodoin for their guidance. I would like to thank my family for their unshaken faith that I would eventually see the light at the end of the tunnel and their loving support. Finally, I would like to thank my classmates Ron and John for their never-ending willingness to provide some tutoring.

Karson A. Sandman

Table of Contents

Abstract.....	iv
Acknowledgments.....	v
List of Figures.....	viii
List of Tables	x
1 Introduction.....	1
1.1 Motivation.....	1
1.2 Problem Statement.....	2
1.3 Sequence of Presentation	3
2 Literature Review.....	4
2.1 Processes for Development.....	4
2.2 Effects of Convection	4
2.3 Development of the Two-Dimensional Convective Cloud Model	7
2.4 Finite Differencing Methods.....	10
2.5 Nuclear Cloud Rise Overview	10
2.6 Existing Modeling Programs	16
2.7 Historical Data From Nevada Test Site	16
3 Methodology.....	18
3.1 Model Development.....	18
3.2 Numerical Methods for Model.....	18
3.3 User-developed Atmospheric Sounding	19
3.4 Initialization	21
3.5 Create Motions in the Model	22
3.6 Weather Data Development.....	22
4 Data Analysis and Discussion.....	24
4.1 Overview.....	24
4.2 Initial Overall Conditions	24
4.3 Model Results for Full Moisture Sounding, 20 K Bubble Temperature.....	27
4.4 Model Results for Full Moisture Sounding, 100 K Bubble Temperature.....	32
4.5 Model Results for Full Moisture Sounding, 1500 K Bubble Temperature.....	36
4.6 Comparison Runs.....	40
5 Conclusions.....	48

5.1	Application to Bomb Burst Modeling.....	48
5.2	Model Summary.....	48
5.3	Suggestions for Further Analysis.....	49
Appendix A: Historical Data (Jodoin, 1994)		52
Appendix B: Model Code		58
Appendix C: METG 612 2D Cloud Modeling Project		76
References.....		96
VITA		97

List of Figures

Figure	Page
2-1: Nuclear Fireball Dynamics (from Bridgman, 2001)	15
4-1: Horizontal Wind Speed vs Elevation, TLH Sounding	25
4-2: Average Pressure vs Elevation, TLH Sounding	26
4-3: Average Actual and Potential Temperature vs Height	27
4-4: Pert Bubble; $\theta = 20$ K, Time = 0 s	28
4-5: Pert Bubble; $\theta = 20$ K, Time = 225 s	29
4-6: Pert Bubble; $\theta = 20$ K, Time = 450 s	30
4-7: Pert Bubble; $\theta = 20$ K, Time = 675 s	31
4-8: Pert Bubble; $\theta = 100$ K, Time = 75 s	33
4-9: Pert Bubble; $\theta = 100$ K, Time = 125 s	34
4-10: Pert Bubble; $\theta = 100$ K, Time 450 s	35
4-11: Pert Bubble; $\theta = 1500$ K, Time 10 s	37
4-12: Pert Bubble; $\theta = 1500$ K, Time 180 s	38
4-13: Pert Bubble; $\theta = 1500$ K, Time 350 s	39
4-14: Shot Dona Ana, $t = 180$ s	41
4-15: Shot Dona Ana, $t = 270$ s	42
4-16: Shot Sanford, $t = 120$ s	43
4-17: Shot Sanford, $t = 270$ s	44
4-18: Shot Koon, $t = 120$ s	45

4-19: Shot Koon, $t = 350s$	46
-----------------------------------	----

List of Tables

Table	Page
2-1: Cloud Top Comparison of Models to Observation (rel to burst point)	17
4-1: Internal Potential Temperature Values, Initial Perturbation $\theta = 20$ K	32
4-2: Internal Potential Temperature Values, Initial Perturbation $\theta = 100$ K	36
4-3: Cloud Top Comparison for Three Shots	47
A-1: Cloud Top Comparison of Models to Observation (rel to burst point)	52
A-2: Cloud Top Comparison of Models to Observation (rel to burst point)	53
A-3: Cloud Top Comparison of Models to Observation (rel to burst point)	54
A-4: Cloud Base Comparison of Models to Observation (rel to burst point)	55
A-5: Cloud Base Comparison of Models to Observation (rel to burst point)	56
A-6: Cloud Base Comparison of Models to Observation (rel to burst point)	57

DEVELOPMENT OF 2-DIMENSIONAL CLOUD RISE MODEL TO ANALYZE INITIAL NUCLEAR CLOUD RISE

1 Introduction

1.1 Motivation

The Department of Defense Land Fallout Interpretive Code (DELFI) is a one-dimensional, physical-empirical model currently used to determine vertical stabilization height for the rise of a nuclear cloud following burst. The model provides the stabilized cloud height and a rudimentary estimate of downwind spread (elliptical divergence of the cloud bubble) to be analyzed subsequently in the Hazard Prediction and Assessment Capability (HPAC) code. The HPAC code utilizes this data to develop a downwind transport model of the effects from the nuclear burst to determine the extent of contamination spread. Due to atmospheric conditions, the one-dimensional model is limited in that the cloud is actually dispersed downwind during the rise. Another significant drawback of a one-dimensional model is the inability to include vertical shear of ambient wind, which is often an important factor in the development and behavior of cloud systems (Rogers and Yau, 1989). The undertaking of this study will be to develop a two-dimensional cloud rise model that is more representative of the actual atmospheric conditions present.

1.2 Problem Statement

The DELFIC cloud rise model (CRM) is a dynamic, one-dimensional, entrainment bubble model of nuclear cloud rise. It consists of a set of coupled ordinary differential equations that represent conservation of momentum, mass, heat and turbulent kinetic energy. The nuclear cloud is defined in terms of: vertical coordinate of its center (the cloud is in some respects treated as a point), cloud volume, average temperature, average turbulent energy density, and the masses of its constituents: air, soil and weapon debris, water vapor and condensed water. Cloud properties and contents are taken to be uniform over the cloud volume. Initial conditions are specified at approximately the time the fireball reaches pressure equilibrium with the atmosphere. Atmospheric conditions (vertical profiles of pressure, temperature and relative humidity) are accepted by the CRM in tabular form (Norment, 1977).

The objective of this study is to create a two-dimensional nuclear cloud rise model that could be used instead of the current one-dimensional cloud rise model of DELFIC. A cloud rise model is developed to model the ascent of convective clouds to equilibrium. This effort involves the numerical solution of a set of partial differential equations to describe the state variables of a two-dimensional cloud. The results are then plotted to give a visualization of the convective cloud rise history for the solved variables. This research explores the feasibility of applying this analysis procedure to nuclear clouds that also rise due to excess buoyancy. A two-dimensional model would provide a much more detailed definition of atmospheric influences including temperature, pressure, vertical wind speed sheer, entrainment, and turbulence within the mushroom

cloud. This would be useful in later studies on particle formation, fractionation, and particle location within the rising/risen cloud.

1.3 Sequence of Presentation

The process to develop and test the two-dimensional convective cloud model follows in Chapter 2. The model is developed primarily from the works presented by Anderson (et al, 1985) simulating the thunderstorm subcloud environment. In Chapter 3, the development of the model is presented to include empirically-based development to ensure workability and subsequent utilization of current weather soundings to test the convective model. Model results are presented in Chapter 4. Model runs are presented for perturbations of 20, 100, and 1500 K. Additionally, three cloud heights are compared to Jodoin's (1994) efforts. Finally, implications of the data comparisons are presented in Chapter 5 to include suggestions for further study.

2 Literature Review

2.1 Processes for Development

A review of the previous effort to develop a cloud rise model is described in Chapter 2. The effects of convection pertaining to cloud rise are explored. The different modeling approaches of “compressible” and “quasi-compressible” models are discussed. Multiple numerical modeling processes are available with various positive and negative attributes. An overview is discussed on appropriate techniques to reach a suitable end product. A procedure to inject the nuclear cloud bubble into the model is postulated.

2.2 Effects of Convection

When a vapor cloud forms, the concentration, sizes, and thermodynamic phase of the hydrometeors are determined by the air motions in and around the clouds together with the characteristics of the aerosol particles that serve as condensation and freezing nuclei. The veering and shearing of the environmental wind often determine the type of convection that can develop, and then exert an influence on the spatial distribution of the precipitation. On the other hand, the microphysical processes of condensation, freezing, melting, and evaporation produce heat sources and sinks, which strongly affect the air circulation. The release of latent heat increases the buoyancy while the drag force of the falling particles causes the opposite effect (Rogers and Yau, 1989). The compilation of these constituents provides a basis for analysis of movement of weather-driven systems through various parts of the atmosphere.

To gain a better understanding of precipitation mechanisms, the microphysical and dynamical processes should be examined as a coupled system. Because of the complexity of the cloud physical processes and the highly nonlinear nature of air motion,

analytic solutions to problems in cloudy convection are extremely rare. Numerical simulation as a tool in the investigation of the interaction between microphysics and dynamics between cloud convection and a larger scale environment has advanced greatly in the last five years (Rogers and Yau, 1989).

Rogers and Yau (1989) present a governing set of equations to account for the dynamic, thermodynamic, and cloud physical process that occur in the numerical analysis of convective processes. To describe convection with precipitation in two dimensions requires at least 7 differential equations: one each for the horizontal and vertical components of air velocity, the continuity equation, the equation for temperature, and one each for water substance in the form of vapor, cloud, and precipitation. These relationships are as follows (Rogers and Yau, 1989):

$$\frac{\partial u_1}{\partial t} = -u_1 \frac{\partial u_1}{\partial x} - u_3 \frac{\partial u_1}{\partial z} - \frac{1}{\rho_0} \frac{\partial \hat{p}}{\partial x} + F_{u_1} \quad (\text{Eq 2-1})$$

$$\frac{\partial u_3}{\partial t} = -u_1 \frac{\partial u_3}{\partial x} - u_3 \frac{\partial u_3}{\partial z} - \frac{1}{\rho_0} \frac{\partial \hat{p}}{\partial x} + g \left(\frac{\hat{T}_v}{T_{v_0}} - \frac{\hat{p}}{p_0} - q \right) + F_{u_3} \quad (\text{Eq 2-2})$$

$$\frac{\partial}{\partial x} (\rho_0 u_1) + \frac{\partial}{\partial z} (\rho_0 u_3) = 0 \quad (\text{Eq 2-3})$$

$$\frac{\partial T}{\partial t} = -u_1 \frac{\partial T}{\partial x} - u_3 \left(\frac{\partial T}{\partial z} + \Gamma \right) + F_T + \phi_T \quad (\text{Eq 2-4})$$

$$\frac{\partial w}{\partial t} = -u_1 \frac{\partial w}{\partial x} - u_3 \frac{\partial w}{\partial z} + F_w + \phi_w \quad (\text{Eq 2-5})$$

$$\frac{\partial \mu}{\partial t} = -u_1 \frac{\partial \mu}{\partial x} - u_3 \frac{\partial \mu}{\partial z} + F_\mu + \phi_\mu \quad (\text{Eq 2-6})$$

$$\frac{\partial R}{\partial t} = -u_1 \frac{\partial R}{\partial x} - u_3 \frac{\partial R}{\partial z} - \frac{1}{\rho_0} \frac{\partial}{\partial z} (\rho_0 R V) + F_R + \phi_R \quad (\text{Eq 2-7})$$

where:

u_1 is horizontal velocity

u_3 is vertical velocity (note that this coincides with w in later reference)

\hat{p} is perturbation pressure

p_0 is the pressure at the elevation evaluated

ρ_0 is density of the air at the elevation evaluated

F_{u_1} is change in horizontal velocity due to turbulent flux

F_{u_3} is change in vertical velocity due to turbulent flux

T is temperature

T_{v_0} is the virtual temperature at elevation evaluated

F_T is change in temperature due to turbulent flux

ϕ_T are the sources and sinks for temperature

q is the mixing ratio of the condensed water

w is the mixing ratio of vapor

μ is the condensation mixing ratio

R is the mixing ratio of the rain

Γ is the lapse rate

ϕ_w are the sources and sinks for the vapor

F_w is the change in vapor due to turbulent flux

ϕ_μ are the sources and sinks for the cloud

F_μ is the change in the cloud ratio due to turbulent flux

V is the fall velocity of the rainwater

ϕ_R are the sources and sinks for the rain

The use of all or some of these equations is dependent upon which modeling approach is utilized.

2.3 Development of the Two-Dimensional Convective Cloud Model

The initial model for this thesis attempts to approximate a solution using the compressible fluid equations. However, some researchers have found that fully compressible equations support sound waves and therefore require much finer time steps for integration (Rogers and Yau, 1989). According to Anderson (et al, 1985) slab-symmetric gust fronts have been analyzed successfully in a two-dimensional plane system using straightforward integration of the full Navier-Stokes equations forward in time.

The full equations for compressible fluid in a two-dimensional non-rotating Cartesian coordinate frame are given by (Anderson, et al, 1985):

$$\frac{Du}{\partial t} = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \mu \nabla^2 u \quad (\text{Eq 2-8})$$

$$\frac{Dw}{\partial t} = -\frac{1}{\rho} \frac{\partial P}{\partial z} - g + \mu \nabla^2 w \quad (\text{Eq 2-9})$$

$$\frac{D\theta}{Dt} = \mu \nabla^2 \theta \quad (\text{Eq 2-10})$$

$$\frac{\partial \rho}{\partial t} = - \left(\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho w)}{\partial z} \right) \quad (\text{Eq 2-11})$$

and

$$P = f(\rho, \theta)$$

where:

u is the horizontal velocity

w is vertical velocity

μ is the atmospheric diffusion coefficient

$f(\rho, \theta)$ represents the state equation for the gas

and the remaining terms have the previously mentioned representation. A disadvantage of this approach is that it gives rise to four linear wave modes (two sound waves and two gravity waves). The sound waves ($c = 350$ m/s) present in this approach require an extremely small time step (less than 0.3 seconds) to ensure computational stability (Anderson et al, 1985).

A comparison follows with a “quasi-compressible” model. This model utilizes the anelastic equations and adds an artificial compressibility term of the continuity equation. The sound waves play the role of bringing the pressure field into dynamic balance and to redistribute the mass field. One example is that sound waves are responsible for moving the warm air in front of the outflow out of the outflow path so the outflow head can progress. In this system, the “pseudo sound” waves travel at speed c_s . This allows for redistribution of mass without the requirement of solving a diagnostic

equation each time step to determine the necessary balance. Choosing the value for c_s is guided by the perpetuation of the gravity waves. The value should be chosen so that c_s is much greater than the speed of the physical signals in the model. Anderson (et al, 1985) found that the quasi-compressible model produced good results as long as the value of c_s was chosen to be at least twice the magnitude as the fastest outflow velocity.

Anderson's (et al, 1985) "quasi-compressible" model is outlined as follows:

$$\frac{Du}{\partial t} = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \mu \nabla^2 u \quad (\text{Eq 2-12})$$

$$\frac{Dw}{\partial t} = -\frac{1}{\rho} \frac{\partial P}{\partial z} + g \frac{(\theta - \bar{\theta})}{\bar{\theta}} + \mu \nabla^2 w \quad (\text{Eq 2-13})$$

$$\frac{D\theta}{Dt} = \mu \nabla^2 (\theta - \bar{\theta}) \quad (\text{Eq 2-14})$$

$$\frac{\partial P}{\partial t} = -c_s^2 \left(\bar{\rho} \frac{\partial u}{\partial x} + \frac{\partial(\bar{\rho} w)}{\partial z} \right) \quad (\text{Eq 2-15})$$

where:

c_s is the speed of propagation for gravity waves passing through the atmosphere.

It has been suggested that this analysis approach would still only resolve a solution to one dimension because Anderson's (et al, 1985) equations initially appear to have only zero-order mean variables. Stull (1988) points out that the parameterization of the turbulent fluxes provides for a first order closure of these equations. This is implemented via a closure approximation specified as gradient transport theory or K-theory. Thus, the use of the diffusion coefficient ensures a coupling in the vertical and horizontal that provides the two-dimensional analysis desired (Stull, 1988).

2.4 Finite Differencing Methods

Finite differencing are utilized for both the spatial and temporal portions of this model. Solutions are developed to comprise “stable” solutions for modeling. Some effects that can cause finite difference models to fail include the development of large amplitude short waves. This is a tendency in the advection equations and in the diffusion part of the model. The Courant-Fredrichs-Lewy (CFL) condition states that the solution of a finite-difference equation must not be independent of the data that determines the solution to the associated partial differential equation. Although the CFL criteria must be met to have a stable system, the criteria in itself does not guarantee stability. The CFL criteria is given for the advection equation by the following:

$$0 \leq c \frac{\Delta t}{\Delta x} \leq \frac{1}{\sqrt{2}} \quad (\text{Eq 2-16})$$

where c is a multiplication factor developed for the advancing of the model system during the numerical integration (Durran, 1999).

2.5 Nuclear Cloud Rise Overview

Norment (1979) developed multiple parameters to represent various properties of the nuclear cloud rise. The nuclear cloud rise is initiated when the fireball reaches an equilibrium pressure with the surrounding atmosphere. The time of initiation was developed from cinefilms of nuclear tests and from radiation measurements of fireballs. Initiation time is given by:

$$t_i = 56t_{2m} W^{-0.30} \quad (\text{Eq 2-17})$$

and the time of the second temperature maximum (surface burst), t_{2m} (seconds) is:

$$t_{2m} = 0.037(1.216^{\lambda/180})W^{(0.49-0.07\lambda/180)} \quad (\text{Eq 2-18})$$

where

λ is scaled height of burst (ft kT^{-1/3})

$$0 \leq \lambda \leq 180$$

W is yield in kilotons (kT)

Initial gas temperature of the fireball, T_i (K), is:

$$T_i = K \left(\frac{t_i}{t_{2m}} \right)^n + 1500 \quad (\text{Eq 2-19})$$

where

$$K = 5980(1.145)^{\lambda/180} W^{(-0.0395+0.264\lambda/180)} \quad (\text{Eq 2-20})$$

and

$$n = -0.4473W^{0.0436} \quad (\text{Eq 2-21})$$

Mass of fallout, m_s (kg), (i.e., soil plus weapon debris) in the cloud at t_i is given

for a surface and above surface bursts with scaled height of burst (ft kT^{-1/3.4}):

$$m_s = k_\Lambda W^{3/3.4} (180 - \Lambda)^2 (360 + \Lambda) \quad (\text{Eq 2-22})$$

$$180 \geq \Lambda \geq 0$$

If Λ is above 180, the burst is considered an air burst in which the fireball doesn't touch the ground and thus no fallout is produced (Norment, 1979).

Constants k_d and k_Λ , 2.182 and 0.07741 (kg ft³) respectively, were determined from an analysis of Teapot Ess fallout. The scaling equation for subsurface bursts is based on Nordyke's scaling function for high-explosive cratering results, and the above-

surface scaling function is based on the volume of intersection of a fireball with the ground (Norment, 1979).

The diameter of the cloud is given by the following relationship:

$$D = 0.65 t^{\frac{3}{7}} W^{\frac{2+\log t}{7}} [km] \quad (\text{Eq 2-23})$$

with W in kT and for time $t \leq 10$ minutes. This formula is not appropriate at late time due to the dominance of the wind shear effect on the cloud width after stabilization. Also, note that low yield (< 50 kT) clouds can easily be much larger than this formula by 10 minutes after burst (DTRA, 2002).

With the aid of Eq 2-26, a temperature can be inserted into the model after sufficient time has passed to accommodate the shortcomings of the Navier-Stokes equations (to handle temperatures at several thousand degrees K). Bridgman (2001) presents an algorithm to couple the properties of temperature, radius and time during the initial diffusion phase directly after burnout. This enables analysis of the relationship between the radius of the bubble with time and temperature. The methodology is as follows:

Given initial conditions, T_{BO} , R_{BO} and t_{BO} and Y_x :

- Arbitrarily decrease T by some delta
- Find cp:

$$c_p^* = \frac{3}{28} (1 + Z^*) R \quad (\text{Eq 2-24})$$

$$Z^* = \frac{\ln T - \ln(2 \times 10^4 \text{ K})}{0.585} \quad (\text{Eq 2-25})$$

- Find R:

$$Y_x = \left(\rho c_p^* T + a T^4 \right) \left(\frac{4}{3} \pi R^3 \right) \quad (R > R_{BO}) \quad (\text{Eq 2-26})$$

- Find the change in radius:

$$\frac{dR}{dt} = \frac{c/4}{1 + \frac{\rho c_p^* T}{a T^4}} \quad (\text{Eq 2-27})$$

- Find time:

$$t = t_{BO} + \int_{R_{BO}}^R \frac{dR}{(dR/dt)} \quad (\text{Eq 2-28})$$

where:

T_{BO} is temperature at burnout

R_{BO} is burnout radius [m] given by the simplified relationship:

$$R_{BO} = \left[\frac{3Y_x}{4\pi(.14)} \right]^{\frac{1}{3}}$$

t_{BO} is time of burnout [s]

Y_x is the x-ray yield ($\sim 70\%$ of total yield)

After pressure equilibrium is reached, the nuclear cloud can be modeled using either two-dimensional hydrocode or by using an analytical solution of bubble gas dynamics to follow the cloud bubble through a still (for modeling purposes) fluid (Jodoin, 1994).

A significant effect of the bubble rise is the vorticity developed which in turn, results in other processes such as entrainment. Vorticity is a local measure of rotation

defined as the curl of the wind velocity (Houze, 1993). Vorticity across the horizontal axis will be predominant in this model versus the coriolis effects of the earth and vorticity effects of an existing wind. The difference in the horizontal wind with increasing height is the most significant cause.

Thus the horizontal component of the vorticity equation is (Houze, 1993) :

$$\omega = \left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \right) j \quad (\text{Eq 2-29})$$

which is captured in the Navier-Stokes model for analysis.

The entrainment that ensues causes mixing that in turn, decreases the buoyancy effects of the rising bubble. Additionally, rising saturated air parcels tend to be diluted by entraining the surrounding environmental air. A balance of air properties follows. An entraining convective cell is less buoyant than a non-entraining cell (Houze, 1993). The entrainment rate can be defined by (Holton, 1992):

$$\lambda = \frac{d(\ln(m))}{dz} \quad (\text{Eq 2-30})$$

then

$$\left(\frac{d(\ln(\theta_e))}{dz} \right)_{cloud} = -\lambda [(\ln(\theta_e))_{cloud} - (\ln(\theta_e))_{environment}] \quad (\text{Eq 2-31})$$

where:

m is the mass of saturated cloud air

θ_e is the equivalent potential temperature

Another sink for the bubble temperature is the mass of fallout itself that is a result of a surface burst. Within the first 24 hours, particulate matter larger than 20 microns in

diameter falls out of the cloud. The fission products themselves only provide approximately 55 grams of mass per kiloton of fission yield and will be considered negligible for the implications of this model. A rule of thumb, the soil mass loading for a surface burst is approximately 0.3 tons of dirt per ton of yield for a 1-MT burst (Bridgman, 2001).

Bridgman (2001) provides a sample development of the fallout cloud for a 1-MT ground burst as shown below. The cloud from the burst reaches neutral buoyancy at approximately 3 minutes at a height of approximately 16 km.

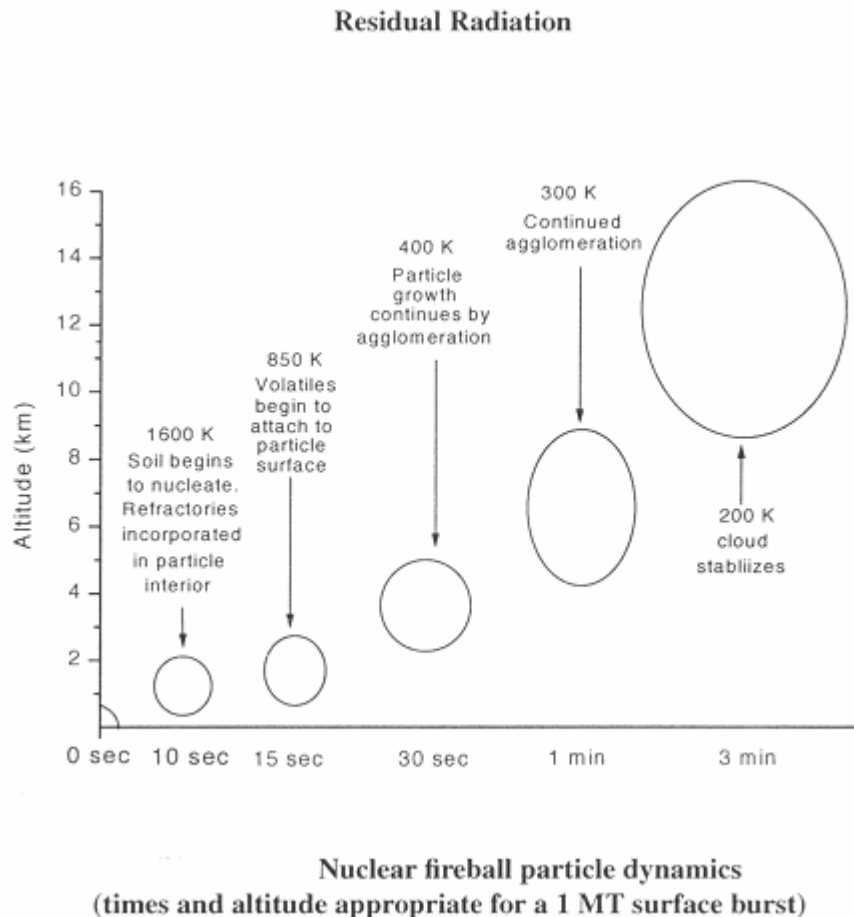


Figure 2-1: Nuclear Fireball Dynamics (from Bridgman, 2001)

2.6 Existing Modeling Programs

The DELFIC CRM is a dynamic, one-dimensional, entrainment bubble model of nuclear cloud rise. The model consists of a set of coupled ordinary differential equations that represent conservation of momentum, mass, heat and turbulent kinetic energy. The nuclear cloud is defined in terms of: vertical coordinate of its center (the cloud is in some respects treated as a point), cloud volume, average temperature, average turbulent energy density, and the masses of its constituents: air, soil and weapon debris, water vapor, and condensed water. The DELFIC model utilizes three different particle size distributions; lognormal (default), power law, and tabular input. For the lognormal distribution, the average particle diameter is 130 μm . Cloud properties and contents are taken to be uniform over the cloud volume. Initial conditions are specified at approximately the time the fireball reaches pressure equilibrium with the atmosphere. Atmospheric conditions (vertical profiles of pressure, temperature and relative humidity) are accepted by the CRM in tabular form (Norment, 1977).

2.7 Historical Data From Nevada Test Site

Historical data exists from nuclear test shots performed at different time periods. The select data listed in Table 2.1 depict relationships for observed cloud top utilizing actual observation, Norment's 1979 model, and corrected and improved versions of the model based on Jodoin's (1994) findings. A more complete listing is detailed in Appendix A.

Table 2-1: Cloud Top Comparison of Models to Observation (rel to burst point)

Shot	Yield (kt)	Observed	Caclulated Cloud Top (m)			Fractional Deviation		
		Cloud Top (m)	1979	Corrected	Improved	1979	Corrected	Improved
Dona Ana	0.037	1940	2831	2802	2796	-0.46	-0.44	-0.44
Sanford	4.9	6530	4946	4986	5942	0.24	0.24	0.09
Koon	110	16150	15549	15713	14995	0.04	0.03	0.07

The models were compared with figures of merit to include Fractional Root Mean Square (FRMS) and Fractional Deviation (FD). The figure of merit for fractional root mean square is (Jodoin, 1994):

$$FRMS = \sqrt{\frac{\sum \left(\frac{z_{rel}^{obs} - z_{rel}^{calc}}{z_{rel}^{obs}} \right)^2}{N}} \quad (Eq\ 2-32)$$

The figure of merit given by fractional mean deviation is calculated as follows (Jodoin, 1994):

$$FMD = \frac{\sum \left(\frac{z_{rel}^{obs} - z_{rel}^{calc}}{z_{rel}^{obs}} \right)}{N} \quad (Eq\ 2-33)$$

3 Methodology

3.1 Model Development

Initially, the model is developed utilizing a user-produced atmospheric sounding. This is necessary in order to ensure an accurately working model with proper controls. Both a dry sounding and a moist sounding are developed and analyzed. The initial base states for the model are developed. A perturbation bubble is created and inserted for analysis. Advection and diffusion are not included in the first model to ensure workability.

As a working model becomes available, various steps are added to the model and tested separately to include mixing and advection of the potential temperature, mixing of the horizontal and vertical winds, advection of the horizontal and vertical winds, and finally, the equations for moisture. Subsequently, a sounding is retrieved from a weather station for analysis. After a working model is developed, controls are inserted into the model to allow for a significant increase in the loading of potential temperature to simulate the aftereffects of an explosion, specifically, the heat bubble produced, for analysis of rise and spread into the atmosphere.

3.2 Numerical Methods for Model

Advection of the model is accomplished using a second-order leap-frog scheme. The pressure gradient and divergence terms are calculated with a forward-backward scheme. At the end of the calculations, an Asselin filter is applied to prevent the leap-frog scheme from separating into two separate solutions (Huffines, 2000). The leapfrog integration is not performed on the potential temperature.

The Asselin leap-frog method is a predictor-corrector method that applies a centered first-order time filter. The time filter is applied as follows (Durran, 1999):

$$\phi^{n,*} = \phi^n + \gamma(\phi^{n-1} - 2\phi^n + \phi^{n+1}) \quad (\text{Eq 3-1})$$

$$\phi^{n-1} = \phi^{n,*} \quad (\text{Eq 3-2})$$

$$\phi^n = \phi^{n+1} \quad (\text{Eq 3-3})$$

3.3 User-developed Atmospheric Sounding

A gridded framework is developed with a 18 km (horizontal) by 18 km (vertical) domain. The domain is adjusted dependent upon the size of the bubble inserted to ensure uniformity of dispersion and to preclude the development of systematic disturbances including gravity-wave induced effects that introduce exponential errors into the solution. A diffusion constant is selected based on known useable model parameters. Arrays are developed for average potential temperature, specific humidity, and horizontal winds. Pressure and density arrays are created using the hydrostatic equation and equations of state. The average potential temperature for a moist sounding is developed from the following equation (Weisman and Klemp, 1982):

$$\bar{\theta}(z) = \begin{cases} \theta_0 + (\theta_{tr} - \theta_0) \left(\frac{z}{z_{tr}} \right)^{\frac{5}{4}} & z \leq z_{tr} \\ \theta_{tr} \exp \left[\frac{g}{c_p T_{tr}} (z - z_{tr}) \right] & z > z_{tr} \end{cases} \quad (\text{Eq 3-4})$$

$$H(z) = \begin{cases} 1 - \frac{3}{4} \left(\frac{z}{z_{tr}} \right)^{\frac{5}{4}} & z \leq z_{tr} \\ .25 & z > z_{tr} \end{cases} \quad (\text{Eq 3-5})$$

$$\overline{q_v^*}(z) = \frac{380}{p(z)} \exp \left[17.27 \frac{(\overline{\pi}(z)\overline{\theta}(z) - 273)}{(\overline{\pi}(z)\overline{\theta}(z) - 36)} \right] H(z) \quad (\text{Eq 3-6})$$

and

$$\overline{q_v}(z) = \min(q_{v_{\max}}, \overline{q_v^*}(z)) \quad (\text{Eq 3-7})$$

where

$\overline{\theta}$ is average elevation potential temperature

$$\theta_0 = 300 \text{ K}$$

$$\theta_{tr} = 343 \text{ K}$$

z is elevation height

$$z_{tr} = 12 \text{ km}$$

$$c_p = 1004 \text{ J/kg-K}$$

$$T_{tr} = 213 \text{ K}$$

$$g = 9.81 \text{ m/s}^2$$

H is humidity

$\overline{\pi}$ is average elevation Exner function value

\bar{p} is average elevation pressure

$$q_{v_{\max}} = 0.014$$

One-dimensional data arrays are developed on the initial sounding data for the following average values: potential temperature, pressure, horizontal wind, and specific humidity. The vertical wind before perturbation is assumed to be zero for this model (hydrostatic balance) due to the overwhelming effects produced by the burst and subsequent motion of the bubble. Additionally, perturbation values are developed for the potential temperature, pressure, horizontal and vertical winds, and specific humidity. Successful output of the model leads to utilization of real weather soundings for model development purposes.

3.4 Initialization

The domain of the problem is configured in the x and z directions. Constants for ideal gas are input and base state variables are developed. The following two-dimensional arrays are initialized: pressure, vapor pressure, temperature, dew point temperature, wind direction, wind speed, density, potential temperature, relative humidity, specific humidity, and the moist gas constant. Two-dimensional variables are initialized using base state information. A dry sounding is utilized. A cold bubble (-10 K) is created in the two-dimensional field. The perturbation pressure field is developed utilizing the hydrostatic equation. Two-dimensional arrays are created for u, w, temperature, and pressure to include the perturbation.

3.5 Create Motions in the Model

The equations of motion described in Chapter 2 are utilized to develop the change in u , w , p , and θ with respect to time for dry model. Mixing, diffusion, divergence, and advection are added. Boundary conditions are created for the model to include no change in horizontal wind speed at the left and right boundaries. Furthermore, change in vertical wind is set to zero at the upper and lower system boundaries, potential temperature is unchanged at the boundaries, as is the rate of change of pressure (assume hydrostatic balance). Motions are created in the model when advection is added. Working model output is compared to verified data from METG 612, 2000 class notes and corrections are made utilizing compressible and ‘quasi-compressible’ fluid modeling techniques and record outputs based on Anderson’s (et al., 1985) work. The model analysis is halted when the convective cloud reaches buoyancy stabilization. (It is assumed that the bubble could overshoot neutral buoyancy height but will stabilize to a neutral buoyancy). Finally, a moist sounding is added to ensure validity of convective rise for this model.

3.6 Weather Data Development

For realistic modeling purposes, a weather sounding is retrieved from the Air Force Weather Agency (AFWA) website to develop the atmospheric conditions that will drive the model. Sounding data points are not equally spaced with elevation and some data fields are left blank. Thus an interpolation program is utilized to provide a equally-spaced, gridded (for numerical analysis) representation of the data from ground level to the highest elevation reading provided from the sounding. A representative text file of the sounding is input into a *MATLAB*-7-based computational program. Small perturbations are to be analyzed initially to determine feasibility of the model.

Subsequent higher temperature perturbations are loaded with appropriate controls in an attempt to reach realistic nuclear explosion temperatures.

4 Data Analysis and Discussion

4.1 Overview

The results of the modeling effort prove effective for smaller input perturbation temperatures while larger perturbations require a time step one to two orders of magnitude smaller. Additionally, changes in the buoyancy wave propagation constant (c_s), and the diffusion constant (K) are required to adjust the model to analyze hotter perturbation inputs. Comparisons are made for input perturbation potential temperatures of 20 K, 100 K, and 1500 K. Vertical rise to buoyancy, vertical velocity, pressure effects, and horizontal wind effects are discussed.

4.2 Initial Overall Conditions

Initially, c_s is set to 75 m/s and K was set to 75 m²/s. A weather sounding from Tallahassee, Florida dated 13 Jul 2003 is utilized and gridded for 100 m vertical and horizontal grid distance. Figure 4-1 depicts the horizontal wind profile in the east-west direction for the sounding on the date noted. Note that the largest magnitudes of wind speed were in the east-west direction but the model could easily be modified to perform analysis in various directions. The profile demonstrates the varying wind profile and causing the shear noted previously. The east-west wind speeds ranged from -3 to 32 m/s.

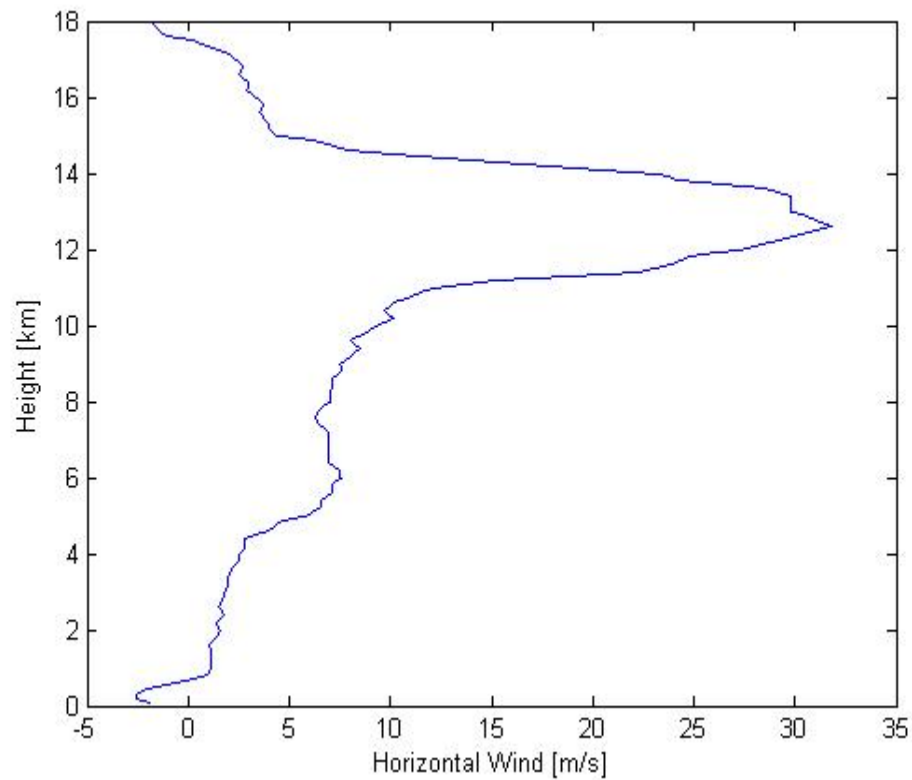


Figure 4-1: Horizontal Wind Speed vs Elevation, TLH Sounding

The pressure varies with height as shown for this sounding. The surface pressure was approximately 1×10^5 Pa and the pressure decreases to 900 Pa at an elevation of 18 km.

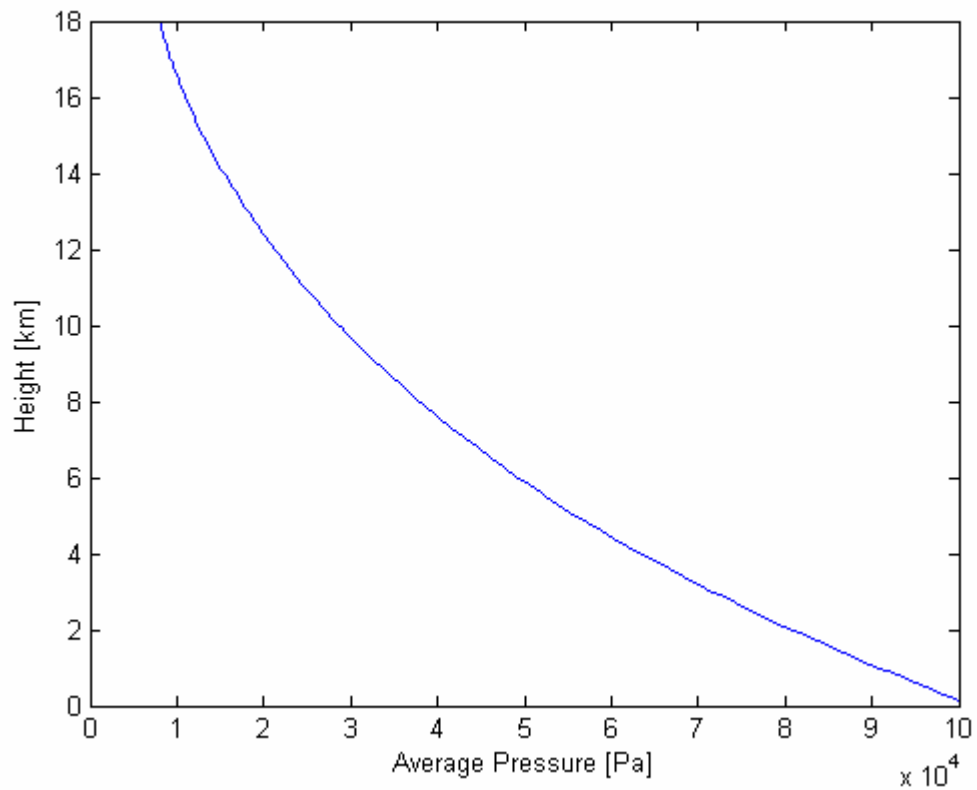


Figure 4-2: Average Pressure vs Elevation, TLH Sounding

Finally, the potential temperature varies with height as depicted (actual temperature is provided for comparison). Note the dramatic change in potential temperature as the elevation changes from the troposphere to the stratosphere around 14 km for this sounding.

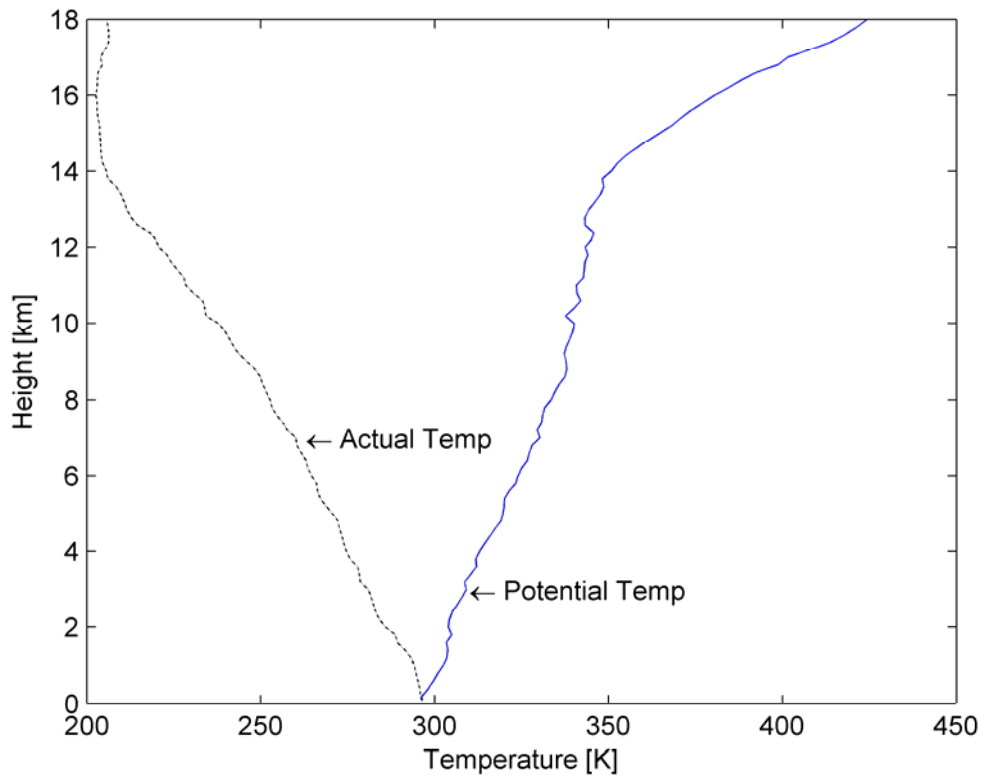


Figure 4-3: Average Actual and Potential Temperature vs Height

4.3 Model Results for Full Moisture Sounding, 20 K Bubble Temperature

A perturbation potential temperature of 20 K was input into the model. The initial bubble produced a perturbation pressure field ranging in values from 0 to 2500 Pa before the model began to run the time step. As shown below, the model initially simulated a bubble that was hottest in the middle and cooled toward the outer radius. The model assumed negligible vertical wind for the first run although the integrations account for any pre-existing activity. No vertical wind was present for the initiation of the model.

The horizontal wind profile depicts the changes in horizontal wind speed with transition to the stratosphere.

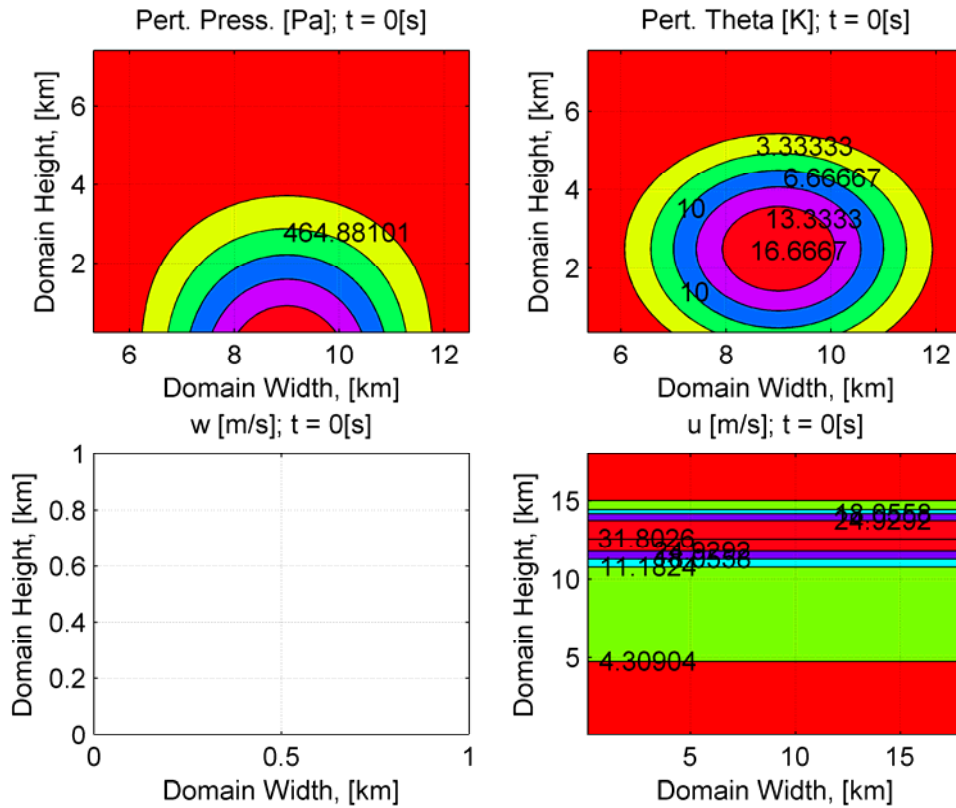


Figure 4-4: Pert Bubble; $\theta = 20$ K, Time = 0 s

The model was run for a total time of 700 s. Interim readings were taken to demonstrate the changes in the model at various points. At 225 s, the perturbation bubble has risen to approximately 7 km. Perturbation pressure has decreased to 1050 Pa. The perturbation potential temperature has cooled to 10 K at the warmest part of the bubble. The vertical wind speed ranged in magnitude to 27 m/s. The horizontal wind profile created a piling effect on the left side of the bubble which was expected as the

predominant winds were coming from the west representing the left side of each of the pictures.

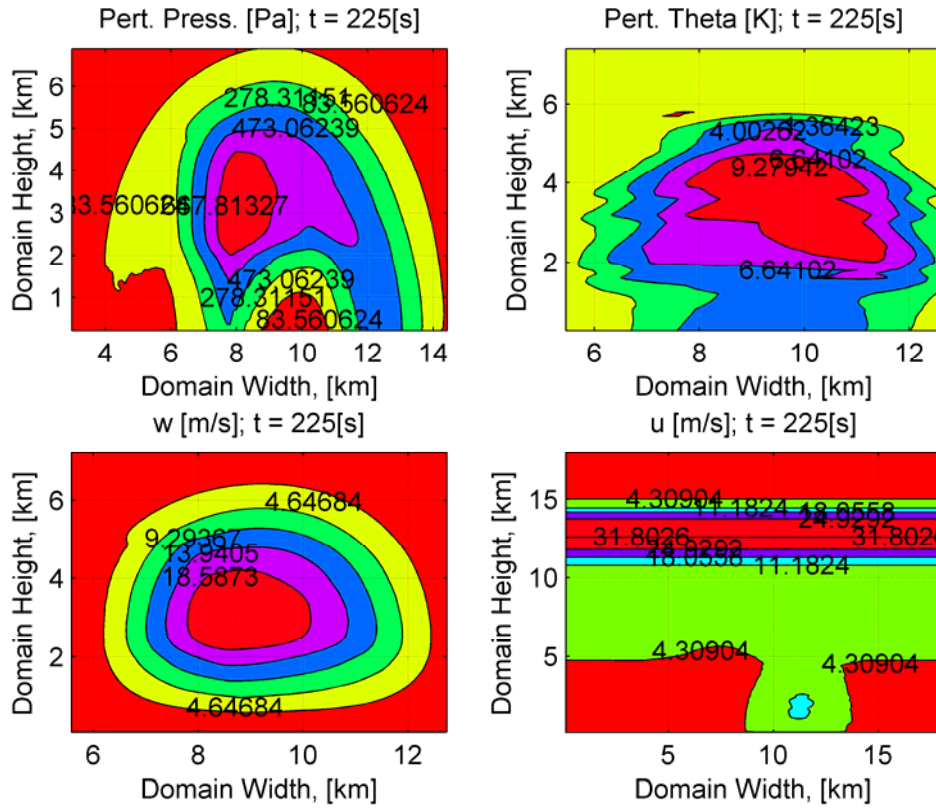


Figure 4-5: Pert Bubble; $\theta = 20$ K, Time = 225 s

At a model time of 450 s, notable diffusion of the bubble has taken place. The perturbation bubble has risen to an elevation of approximately 8 km. Perturbation pressures have decreased to approximately 1020 Pa. The bubble temperature has cooled to approximately 5 K. Vertical wind speed ranges to 15 m/s. The horizontal wind profile has created a “void effect” to keep the mass balance in the system around the bubble.

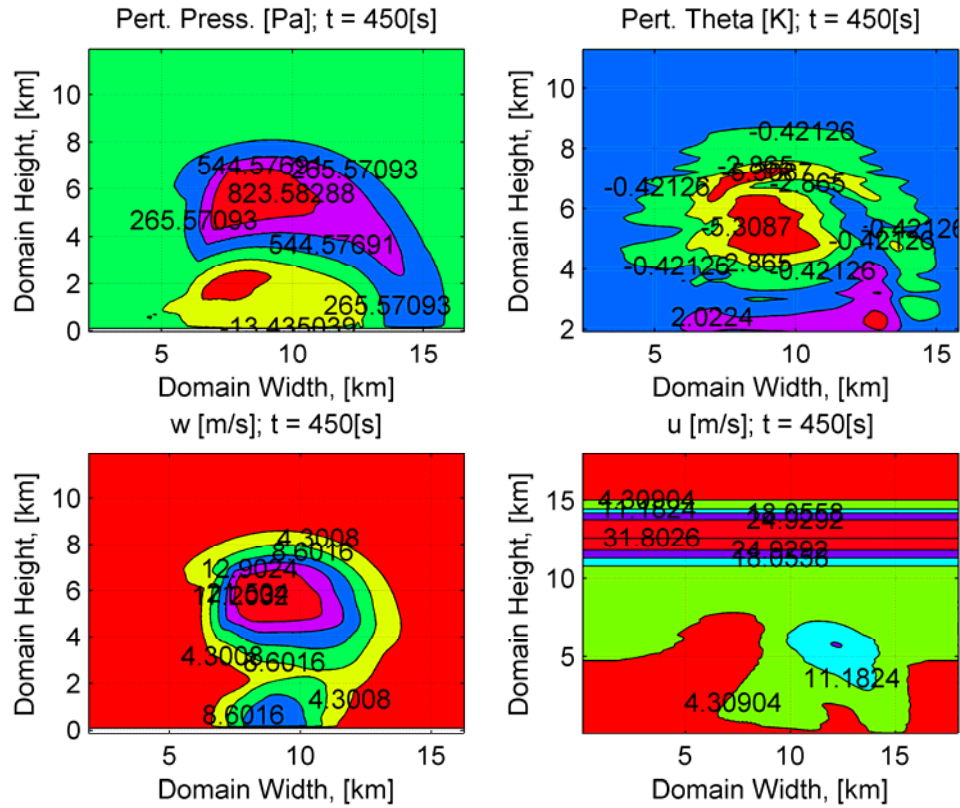


Figure 4-6: Pert Bubble; $\theta = 20$ K, Time = 450 s

A final readout was taken at a model run time of 675 s. The bubble's vertical ascent had slowed enough to be considered buoyant at approximately 10 km and had continued to diffuse in the horizontal direction with only a slight increase in elevation. Perturbation pressure had decreased to approximately 150 Pa. The bubble potential temperature had cooled to approximately 3 K and some of the values had gone negative as a result of the surrounding atmospheric potential temperatures. Overall vertical winds decreased but values still ranged up to 16 m/s.

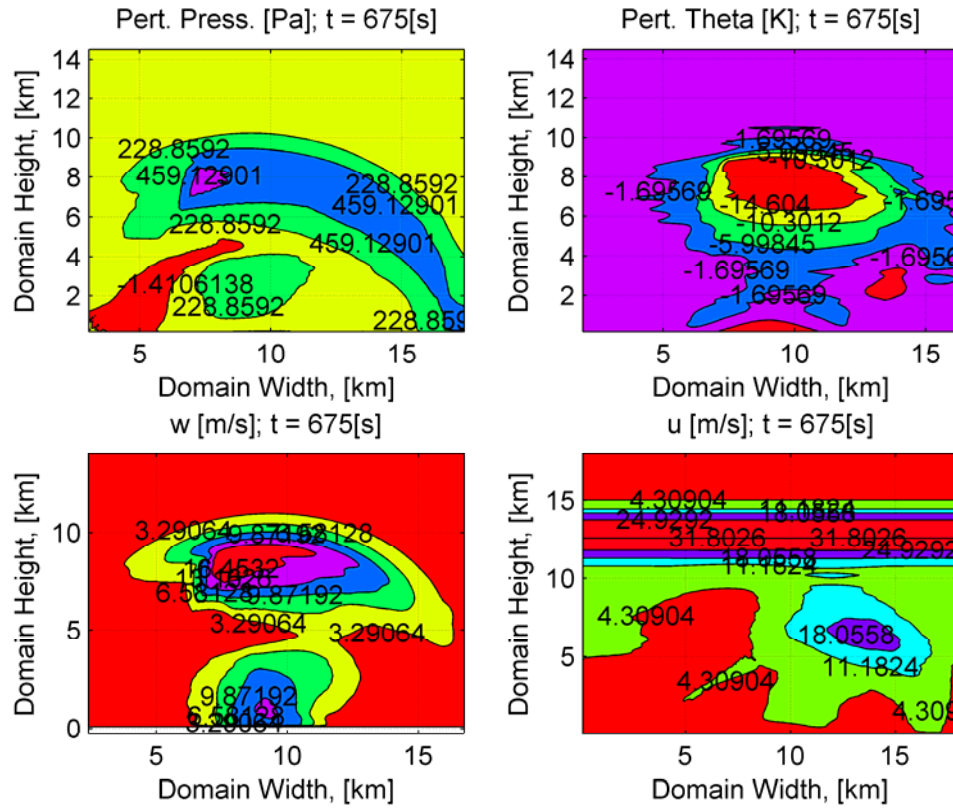


Figure 4-7: Pert Bubble; $\theta = 20$ K, Time = 675 s

An overall snapshot of the perturbation temperatures is provided below. Results are shown are for elapsed model run time at the noted elevations. The time-phased rate of change for the temperature was larger for the higher temperature difference as noted below.

Table 4-1: Internal Potential Temp. Values, Initial Perturbation $\theta = 20$ K

		Horizontal Location [km]						
Model Time (s)	Elevation [km]	8	8.1	8.2	8.3	8.4	8.5	8.6
(150 s)	4.1	13.99	14.34	14.66	14.93	15.18	15.40	15.58
	4.2	13.64	14.04	14.40	14.72	15.00	15.25	15.45
	4.3	12.96	13.42	13.84	14.20	14.53	14.81	15.06
(275s)	4.1	7.39	7.46	7.53	7.60	7.67	7.73	7.79
	4.2	7.25	7.26	7.29	7.32	7.36	7.40	7.45
	4.3	7.39	7.34	7.30	7.27	7.26	7.26	7.27
(400 s)	4.1	6.60	6.70	6.79	6.88	6.95	7.02	7.08
	4.2	6.11	6.19	6.28	6.36	6.44	6.51	6.57
	4.3	5.58	5.64	5.71	5.79	5.86	5.92	5.98

4.4 Model Results for Full Moisture Sounding, 100 K Bubble Temperature

Another run was made in the model for a bubble temperature of 100 K. Parameters c_s and K were set to 100 for each. The same initial settings described in section 4.2 were utilized. A smaller time step of .8 seconds was implemented. The initial conditions of the model were a perturbation with a radius of 1200 m located at an elevation of 1600 m. A reading was taken at $t = 75$ s as shown below. A mushroom effect is noted in the pressure field and the horizontal wind is experiencing a significant rollover effect. The cloud had risen to an elevation of 2.8 km.

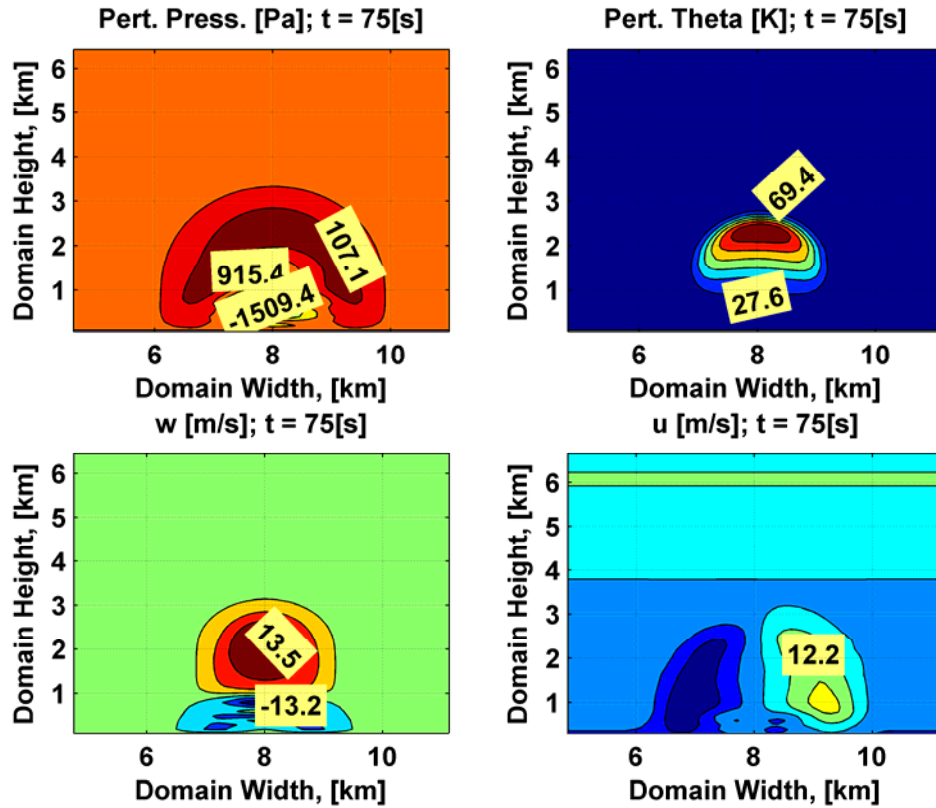


Figure 4-8: Pert Bubble; $\theta = 100$ K, Time = 75 s

Again, a reading was taken at a run time of 125 s. The bubble had risen to an elevation of approximately 3.2 km. Perturbation pressures decreased and ranged to 2250 Pa. The bubble perturbation temperature had diffused and cooled but some small areas in the cloud still reached temperatures of 99 K for the perturbation. The vertical wind speeds ranged from -20 m/s to 35 m/s in the model. The horizontal wind profile began to move over the top of the bubble.

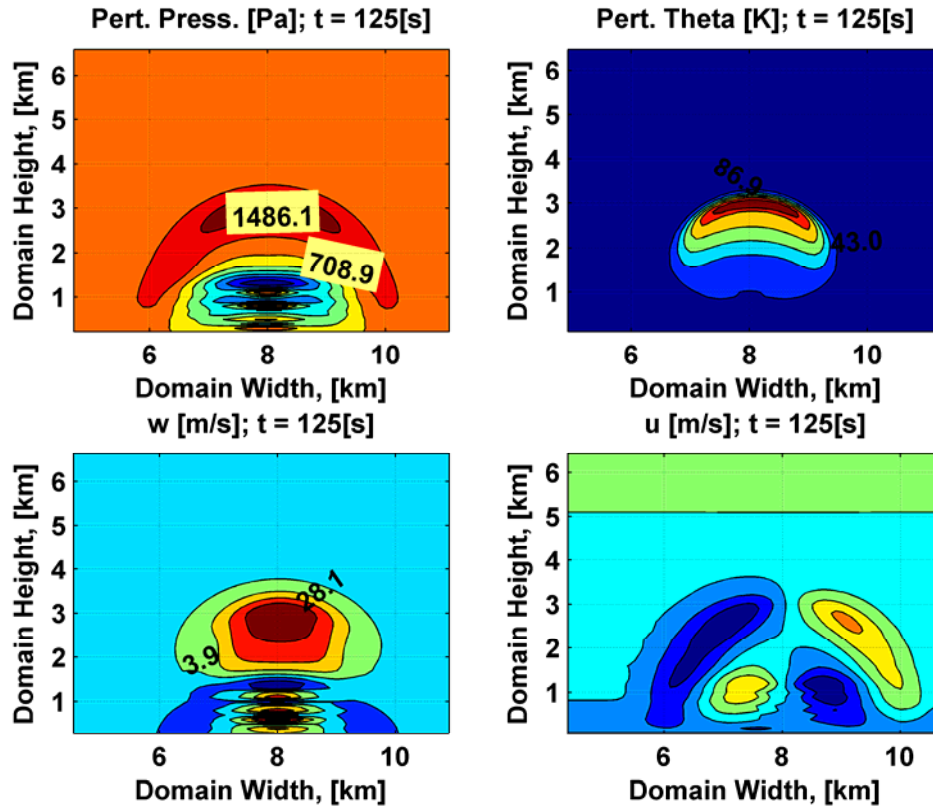


Figure 4-9: Pert Bubble; $\theta = 100$ K, Time = 125 s

At a time of 225 s, the model depicted the following outputs. The thermal bubble had significantly diffused and had reached an elevation of 4.5 km. The horizontal diffusion spanned over 4 km. Vertical velocities still had small pockets of high values but overall had seen a significant decrease.

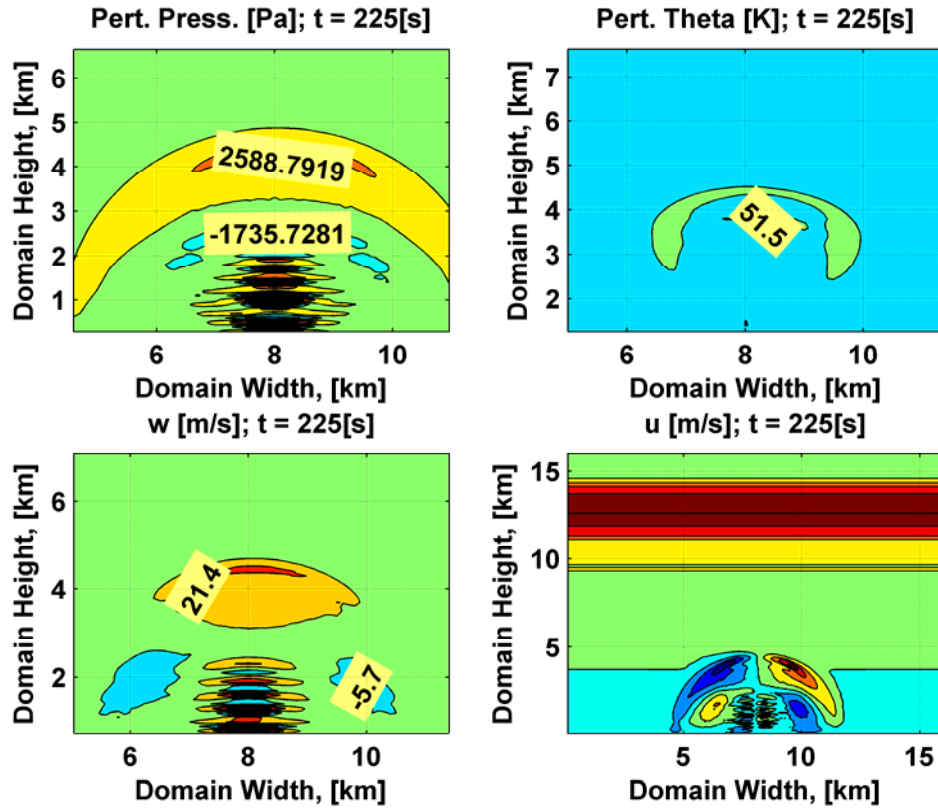


Figure 4-10: Pert Bubble; $\theta = 100$ K, Time 450 s

At model run time of 250 s, the cloud did begin to show some sign of gravity waves but also had significantly diffused to consider it reaching a buoyancy point.

The following table depicts the time rate of change of segments of the model for times shown in the above discussion. Note that the potential temperature values go negative as the perturbation continues to neutral buoyancy.

Table 4-2: Internal Potential Temp. Values, Initial Perturbation $\theta = 100$ K

Model Time (s)	Elevation [km]	Horizontal Location [km]						
		8	8.1	8.2	8.3	8.4	8.5	8.6
75	2.5	96.29	97.12	97.35	96.94	95.42	91.83	84.73
	2.6	94.57	96.53	96.65	94.89	90.70	83.20	71.56
	2.7	79.29	82.45	82.47	79.31	72.71	62.52	49.21
125	2.5	53.78	53.34	53.36	53.84	54.78	56.20	58.05
	2.6	59.32	58.98	59.03	59.51	60.38	61.52	62.74
	2.7	62.84	62.91	63.05	63.27	63.55	63.86	64.37
225	2.5	0.96	-3.76	-2.67	3.66	12.53	20.81	26.45
	2.6	23.00	21.47	21.55	23.29	25.55	27.56	29.28
	2.7	27.31	26.78	26.61	26.73	27.15	27.85	28.87

4.5 Model Results for Full Moisture Sounding, 1500 K Bubble Temperature

Finally, analysis was performed on a 1500 K perturbation bubble. A 0.12 s time step was utilized and the model was run for 500 s. The constant K was set at $100 \text{ m}^2/\text{s}$ and c_s was set to 100 m/s . The bubble was input at an elevation of 1500 m with a radius of 800 m. The sounding was the Tallahassee-based weather input. The model had some breakdown as seen in the 100 K perturbation such as buoyancy effects and frequency inflections in the numerical calculations. In this run, the bubble was inserted as a uniform bubble assuming rapid growth with negligible diffusion gradient of heat from bubble surface to center. Additionally, the perturbation pressure was developed after running the calculation—not as an initial condition. After 10 s, the potential temperature remained the same with limited edge diffusion; perturbation pressure ranged to 1450 Pa; vertical winds ranged to 60 m/s; and the horizontal winds were affected as shown below.

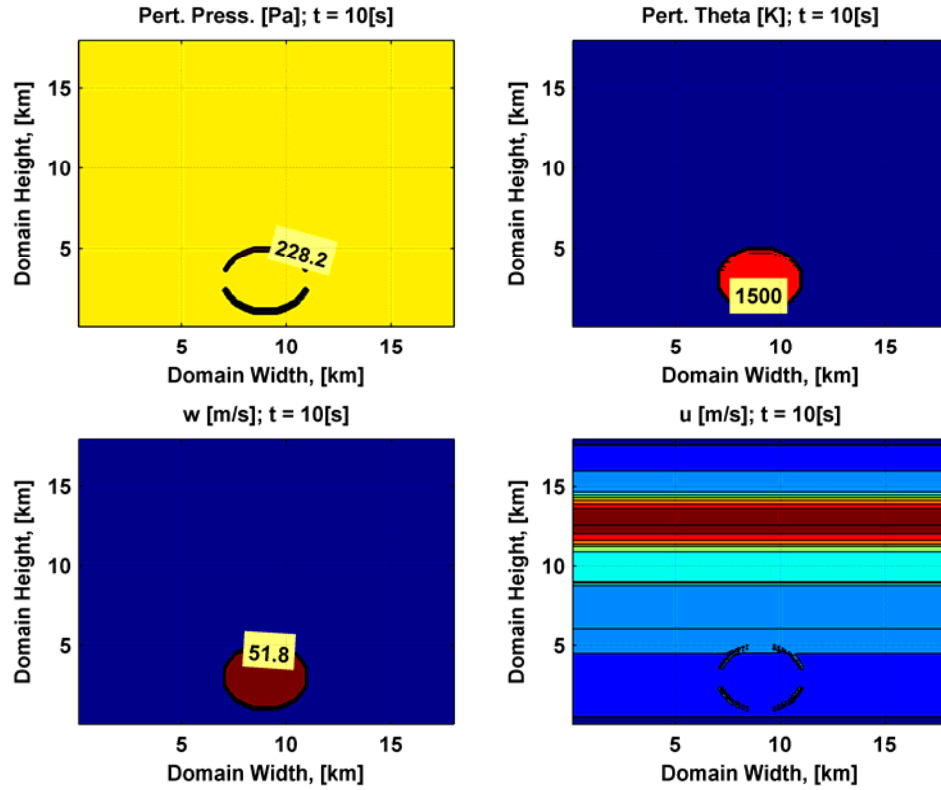


Figure 4-11: Pert Bubble; $\theta = 1500$ K, Time 10 s

At 180 s, the model had diffused somewhat and experienced some horizontal distribution. The bubble had risen to 5 km (cloud top). Perturbation pressures ranged to 10 kPa. Potential temperature ranged to 1200 K. Vertical velocities reached a magnitude of 135 m/s at spurious locations. The horizontal winds experienced a turning effect.

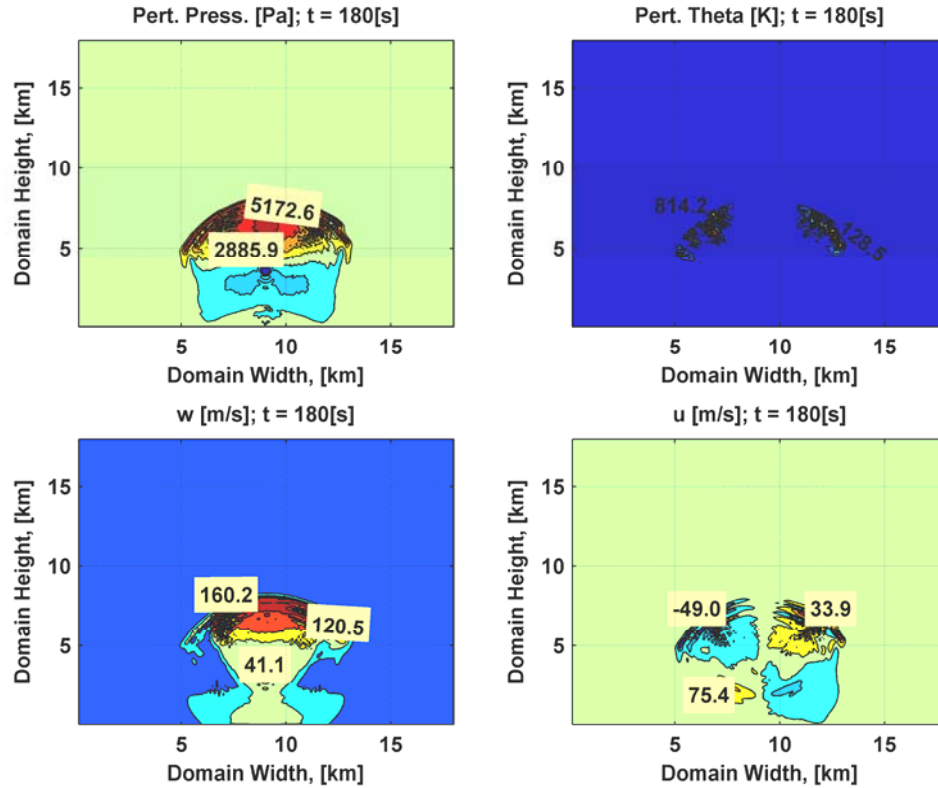


Figure 4-12: Pert Bubble; $\theta = 1500$ K, Time 180 s

At 350 s, the bubble had dissipated in the horizontal domain significantly with a spread of over 10 km. Perturbation pressured were reduced ranging to 7000 Pa. Overall temperature decreased but small volumetric pockets still existed with temperatures close to the initial perturbation temperature. Vertical velocities ranged to 200 m/s in isolated locations while most of the velocities were less than 100 m/s. Significant rollover was noted for the horizontal wind.

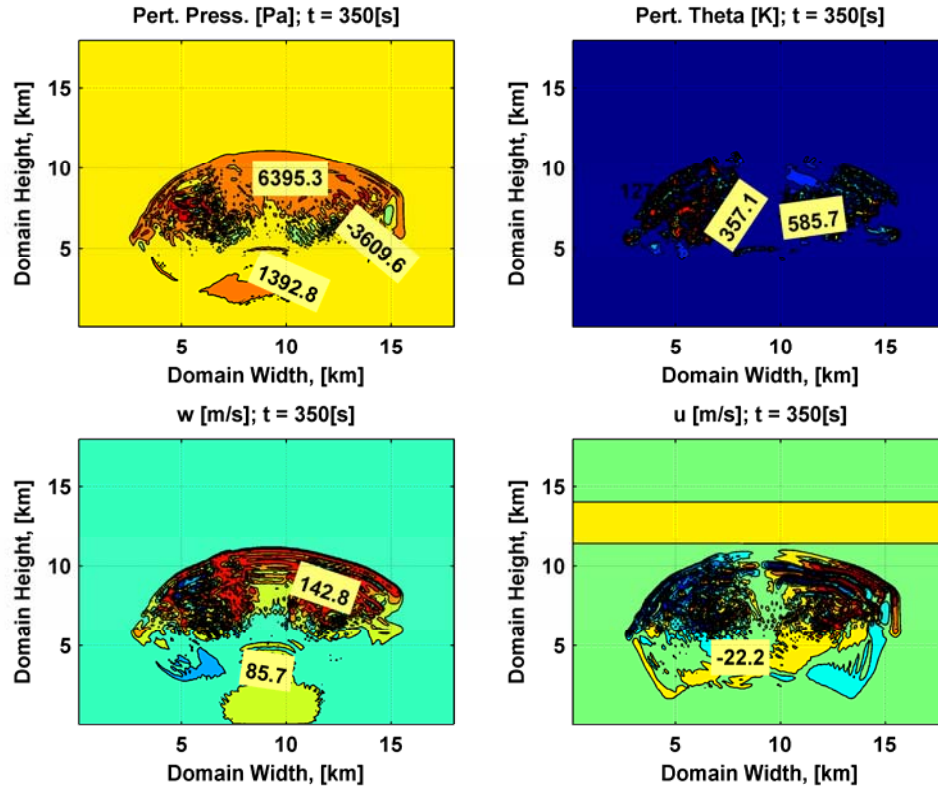


Figure 4-13: Pert Bubble; $\theta = 1500$ K, Time 350 s

The model was stopped at a run time of 500 s. At that time, the perturbation cloud had reached a vertical height of 12 km and had spread approximately 15 km in the horizontal domain.

According to Glasstone (1977), vertical velocities of the nuclear cloud for a 1-MT air burst (based on test data analysis) can reach 150 m/s at lower elevations (3 km) and slow as the cloud reaches a buoyancy height of 20 km. The author also notes that the values are rough averages that can deviate immensely when differing weather conditions apply. Glasstone also presents a time-vs-cloud height relationship for a low altitude burst

with a 1-MT yield. Initially, the cloud experiences a rapid rise for the first 4 minutes after explosion and slowing to an approximate buoyancy point of approximately 20 km after 6 minutes.

In comparison, this model ran for 500 s starting at an elevation of 1.5 km and was still moving in the vertical direction at an elevation of 12 km when the run time ended.

4.6 Comparison Runs

After finally developing a model that could pass high temperatures for extended periods of time, a cloud rise analysis was performed against some of shots listed in Appendix A. Overall, the findings showed significantly less cloud rise than the calculated or observed values. Some of the results are as shown.

At model run time of 180 s, Shot Dona Ana had a horizontal distribution of approximately 1.5 km and had risen to a height of 1.8 km in the temperature plot. A small amount of horizontal shear and mixing was noted.

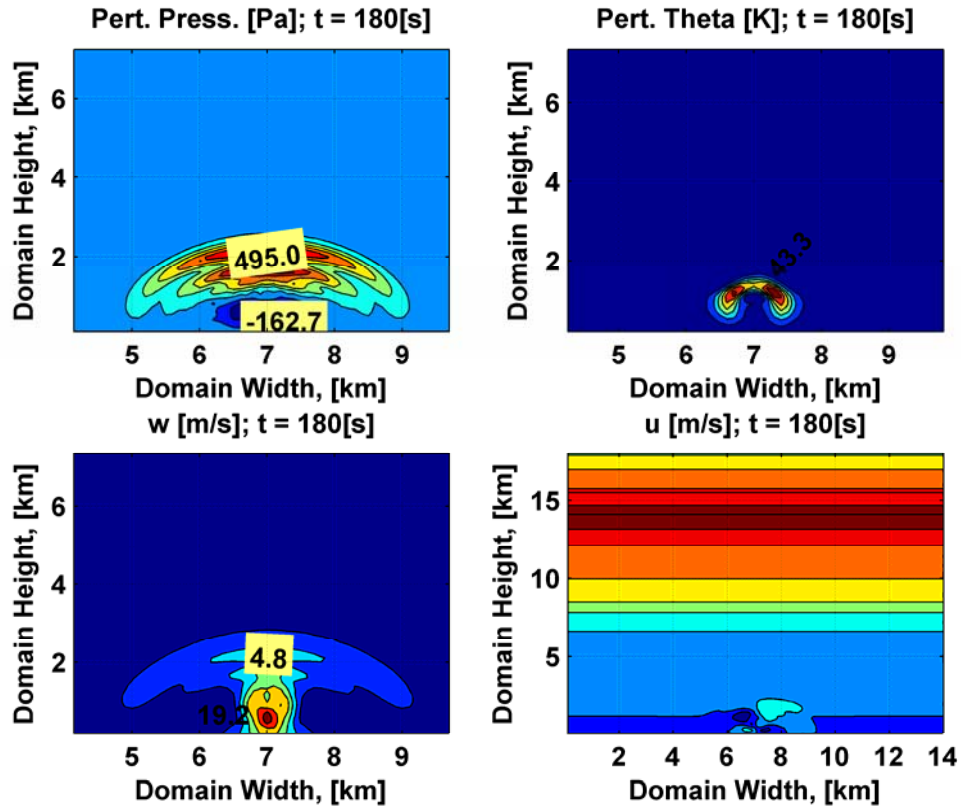


Figure 4-14: Shot Dona Ana, $t = 180$ s

At model run time of 270 s, the horizontal distribution spread over a two km domain and the cloud had risen to 2.1 km in the temperature plot. The cloud rose to a height of approximately 3.4 km before dissipating at a model run time of 350 s. The cloud divergence extended over 3 km when the dissipation occurred.

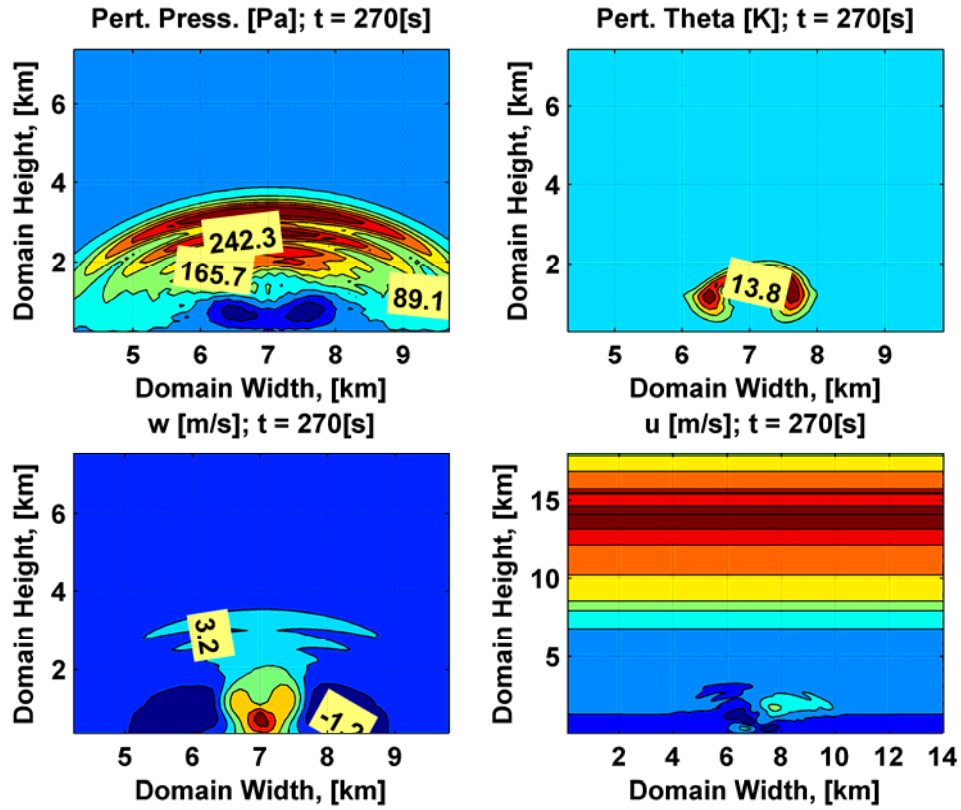


Figure 4-15: Shot Dona Ana, $t = 270$ s

Shot Sanford had significant divergence of the temperature field at a model run time of 120 s. At this point, the bubble spanned over 2.5 km in the horizontal domain. The bubble reached an elevation of 2.2 km.

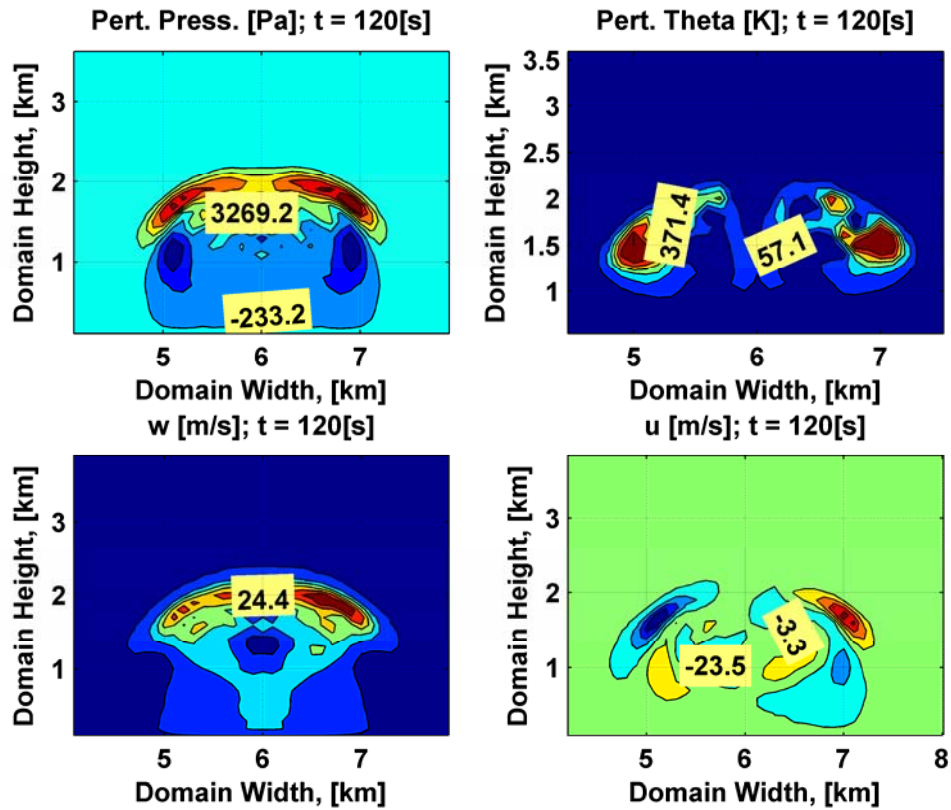


Figure 4-16: Shot Sanford, $t = 120$ s

Shot Sanford rose to a height of over 5.1 km by model run time of 270 s as shown below. Several boundary layer errors were introduced into the model with values increasing exponentially accuracy is questionable.

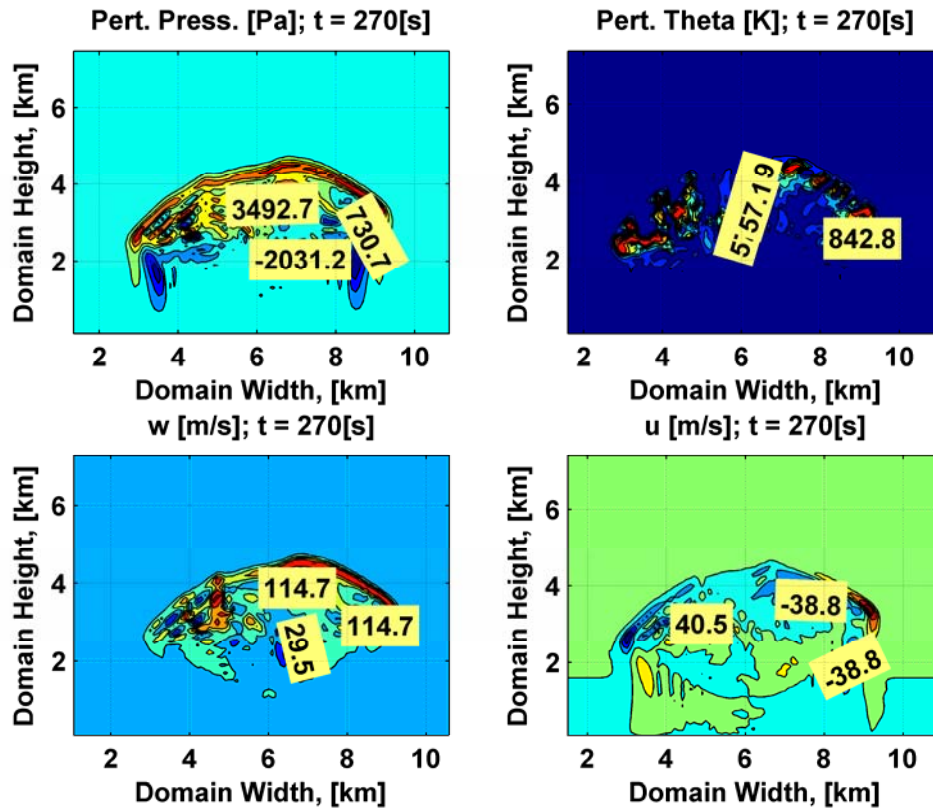


Figure 4-17: Shot Sanford, $t = 270$ s

Finally, shot Koon was analyzed in the model as shown below. The cloud ascended rapidly during the first two minutes of run time. The bubble showed significant diffusion and breakup at this point.

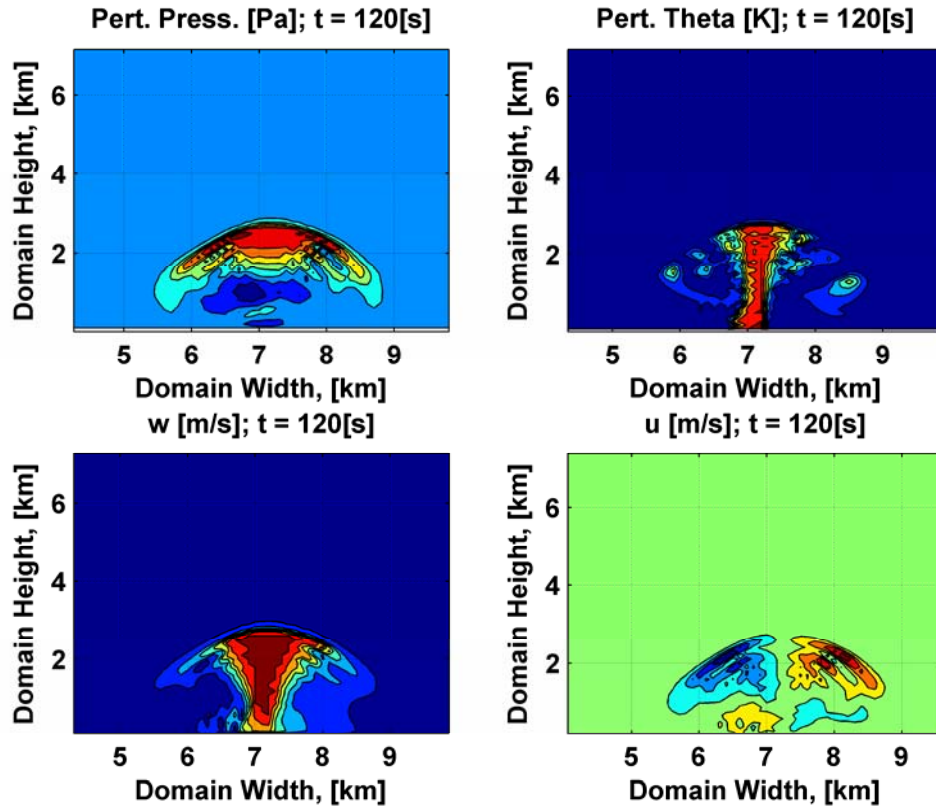


Figure 4-18: Shot Koon, $t = 120$ s

Again, exponential errors next to the boundary layers introduced significant error into the analysis after a model run time of approximately 240 s. At a model run time of 350 s, the cloud was still rising and had reached a height of 12.1 km. The cloud had a horizontal span of 14 km by this time.

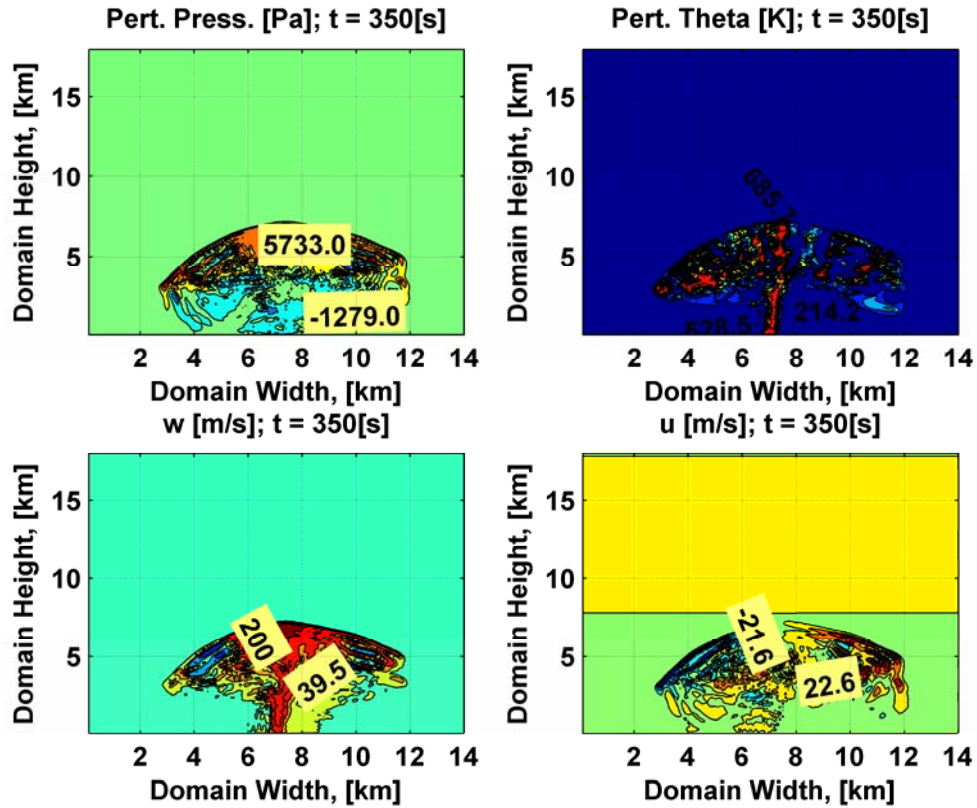


Figure 4-19: Shot Koon, $t = 350$ s

In comparison with the previous data, the model runs for three shots listed above were significantly smaller for the cloud top height. However, the model had not reached complete neutral buoyancy by the end of the model runs.

Table 4-3: Cloud Top Comparison for Three Shots

Shot	Yield (kT)	Observed Cloud Top (m)	Calculated Cloud Top (m)			
			1979	Corrected	Improved	Convective
						Cloud Rise Model
Dona Ana	0.037	1940	2831	2802	2796	3400
Sanford	4.9	6530	4946	4986	5942	5100
Koon	110	16150	15549	15713	14995	12100

Some of the factors that influence these results include the model runs introducing significant boundary errors around the bubble after approximately 3.5 to 4 minutes of run time. Additionally, the perturbation was shown not to have cooled to the point of equilibrium and was traveling in the horizontal direction. The rise times for this model were considerably longer than what was expected. Not including effects of momentum when the perturbation is put into the model could play a significant role.

5 Conclusions

5.1 Application to Bomb Burst Modeling

This model provides insight into the internal processes occurring within the turbulent mixing experienced during the rise of a hot perturbation bubble analogous to a nuclear bubble rise after several seconds of initial growth (and decrease in temperature) as shown in Figure 2-1. The model represented the developed hot gas cloud rise resulting from a nuclear explosion to further develop insight into the physics and dynamics of the nuclear cloud bubble rise.

5.2 Model Summary

Initially, a numerical model was built to perform analysis on a hot gas bubble in the atmosphere in order to compare atmospheric effects due to wind shear coupled with known weather conditions on a two-dimensional nuclear bubble as it rises through the atmosphere. The model was tested with 20 K, 100 K, and 1500 K initial perturbation temperatures. Results were reasonable for perturbation temperatures less than 100 K and commensurate with Huffines (2000) findings. The model gave an estimation of the wind shear effects in two dimensions unlike previous models. As the perturbation temperature input into the model increased, however, buoyancy waves developed in the system after 3-4 minutes of analysis causing the model to rapidly accelerate value differencing and thus causing significantly large instabilities.

The model was then compared to Jodoin's (1994) findings. This model produced cloud stabilization heights that were lower than previous analysis (note that Shots Sanford and Koon hadn't reached buoyancy). However, it is not expected that this two-

dimensional analysis should reach a completely comparable correlation to the previously found data but is developed to provide insight into the microphysical processes such as horizontal and vertical wind change, pressure change, and temperature change that are occurring with the rise of the bubble.

Overall, the model did provide detailed insight into the wind shear effects on the hot bubble perturbation. Additionally, the model allows the input of an actual weather sounding and develops two-dimensional output.

5.3 Suggestions for Further Analysis

This model contains multiple variables that need further analysis to better represent the overall cloud rise from the two-dimensional perspective. Numerical analysis is performed using a first order method (Asselin-Leapfrog). Although Durran (1999) comments that the spatial derivative error is usually more significant than the temporal and might not warrant more than a first order solution, it is possible that a few of the methods Durran lists for second or third order accuracy might be better suited for the amplification error detected while running this model. These include buoyancy oscillations that exponentially grow values (or decrease them) to the extent where the perturbation disappears (this occurs with perturbations above 100 K). Additionally, an Asselin filter factor of .09 is used for the model. Durran recommends values ranging to .25 for the faster convective processes. However, higher values result in failure in this model. Multiple analysis runs are required to determine a better value fit for the model and possibly determine which weather conditions result in better applicability.

As in many numerical analysis problems involving numerical integration processes, grid size also plays a significant role. Initially, this model is comprised of

200-m grid spacing in the horizontal and vertical directions. Results are acceptable for the initial “empirical” sounding developed to test the model. However, when the Tallahassee sounding is run with the same grid spacing, the shear effects of the wind quickly cause failure. Although radio-sonde data usually does not produce weather data on such a small scale, interpolation based on simple mathematical application and basic knowledge of weather phenomena in the troposphere and stratosphere can be accomplished to improve input data.

As previously mentioned, buoyancy waves developed in model runs with perturbation temperatures above 100 K. Further study of mass distribution based on the physical conditions of the atmosphere to include the temperature, water vapor in the air, lapse conditions, wind speed, and the conditions of the perturbation bubble introduced into the system could improve the efficiency of the analysis. The value of c_s must be optimized for the model application. Several test runs are made with the model employing varying values of c_s from 10 to $200 \text{ m}^2/\text{s}$. Anderson (et al, 1985) developed a relationship for c_s vs outflow of the perturbation field implying that the value works well when it has at least twice the magnitude of the larger outflow velocities found. However, that system is based on representing convective cloud movement with perturbation temperatures two to three orders of magnitude smaller than this model.

The diffusion coefficient, K, requires further review to ensure the proper working relationship for the analysis performed. According to Stull (1988), values vary for this type of slab symmetric analysis ranging in values from less than 1 to greater than 2000 m^2/s . Additionally, another multiplication factor can be assigned to the potential

temperature diffusion coefficient K versus for that of the other diffusion coefficients for wind and moisture. A differential analysis should be performed to develop a more direct correlation with the application.

The mass loading of a surface burst and subsequent fallout of large particulate matter ($>> 20 \mu m$) removes a significant amount of energy from the cloud. This model did not include the effects of the mass unloading and the subsequent removal of the energy accompanying this effect. Developing a differential time phased fallout variable would be beneficial in determining overall fallout span and would contribute to more effective cloud rise analysis especially loss of heat in the cloud due to soil condensation and removal from the analyzed system. Radiative cooling dominates at the extremely high temperatures (on the order of 10^7 K) created during the burst and should be incorporated into the energy calculation (Bridgman, 2001). Additionally, momentum should be input into the system when the model begins to run.

Thus, with some overall better fits for the coefficient values, the author feels that this model will prove useful in determining internal microphysical attributes not previously seen in the analysis of nuclear bubble rise.

Appendix A: Historical Data (Jodoin, 1994)

Table A-1: Cloud Top Comparison of Models to Observation (rel to burst point)

Shot	Yield (kT)	Observed Cloud Top (m)	Calculated Cloud Top (m)			Fractional Deviation		
			1979	Corrected	Improved	1979	Corrected	Improved
Humboldt	0.0078	1050	759 106	827	1038	0.28	0.21	0.01
Catron	0.021	1344	1 228	1170	1415	0.21	0.13	-0.05
Vesta	0.024	1760	2 283	2294	2247	-0.30	-0.30	-0.28
DonaAna	0.037	1940	1 217	2802	2796	-0.46	-0.44	-0.44
Hidalgo	0.077	2267	2 154	2258	2515	0.04	0.00	-0.11
Quay	0.079	1722	8 236	1543	1768	0.10	0.10	-0.03
Eddy	0.083	1925	8 189	2173	2635	-0.23	-0.11	-0.37
RioArriba	0.09	2870	5 162	1975	2559	0.34	0.31	0.11
Wrangell	0.115	1653	7 417	1640	1861	0.02	0.01	-0.13
Franklin	0.14	3772	5 337	4253	4179	-0.11	-0.13	-0.11
Wheeler	0.197	3740	4 212	3396	3684	0.10	0.09	0.01
Ray	0.2	2644	1 283	2141	2524	0.20	0.19	0.05
Ruth	0.2	2833	8 257	2946	3051	0.00	-0.04	-0.08
JohnnieBoy	0.5	3612	5 470	2565	2818	0.29	0.29	0.22
Laplace	1	4592	9 355	4736	4819	-0.03	-0.03	-0.05
SantaFe	1.3	3753	0 393	3569	4254	0.05	0.05	-0.13
Lea	1.4	3449	1 433	3925	4536	-0.14	-0.14	-0.32
Mora	2	3906	9 419	4374	4521	-0.11	-0.12	-0.16
John	2	6008	7	4197	4540	0.30	0.30	0.24

Table A-2: Cloud Top Comparison of Models to Observation (rel to burst point)

Shot	Yield (kT)	Observed Cloud Top (m)	Calculated Cloud Top (m)			Fractional Deviation		
			1979	Corrected	Improved	1979	Corrected	Improved
DeBaca	2.2	3601	3878	3811	4989	-0.08	-0.06	-0.39
Franklin Prime	4.7	8249	5467	5480	5841	0.34	0.34	0.29
Sanford	4.9	6530	4946	4986	5942	0.24	0.24	0.09
Socorro	6	6207	5776	5792	6287	0.07	0.07	-0.01
Morgan	8	10755	6374	6379	6773	0.41	0.41	0.37
Owens	9.7	9231	8033	7999	7407	0.13	0.13	0.20
Wilson	10	9226	6409	6389	7987	0.31	0.31	0.13
Kepler	10	7069	7612	7611	7648	-0.08	-0.08	-0.08
Fizeau	11	10811	7780	7833	7915	0.28	0.28	0.27
Galileo	11	9830	7554	7573	8029	0.23	0.23	0.18
Doppler	11	9836	7731	7685	7307	0.21	0.22	0.26
Dixie	11	10654	8099	8092	8681	0.24	0.24	0.19
Boltzman	12	8615	10524	10517	9887	-0.22	-0.22	-0.15
Newton	12	8021	7958	8008	8057	0.01	0.00	0.00
Charleston	12	8012	6779	6823	6801	0.15	0.15	0.15
Grable	15	9570	6164	6147	6425	0.36	0.04	0.33
Annie	16	11178	9365	9480	10039	0.16	0.15	0.10
Shasta	17	8264	9347	9215	8717	-0.13	-0.12	-0.05

Table A-3: Cloud Top Comparison of Models to Observation (rel to burst point)

Shot	Yield (kT)	Observed Cloud Top (m)	Calculated Cloud Top (m)			Fractional Deviation		
			1979	Corrected	Improved	1979	Corrected	Improved
Diablo	17	8239	8884	8837	9171	-0.08	-0.07	-0.11
Whitney	19	7624	8459	8474	9042	-0.11	-0.11	-0.19
Stokes	19	9545	8494	8497	8732	0.11	0.11	0.09
Badger	23	9513	7554	7568	8897	0.21	0.20	0.06
Nancy	24	11244	8807	8823	9217	0.22	0.22	18.00
Encore	27	11125	8908	8911	9246	0.2	0.20	0.17
Harry	32	11642	11844	11997	12640	-0.02	-0.03	-0.09
Priscilla	37	11955	10782	10774	11150	0.10	0.10	0.07
Lacrosse	40	11582	7410	9164	8983	0.36	0.21	0.22
Simon	43	12028	12140	12183	12181	-0.01	-0.01	-0.01
Smoky	44	10004	11290	11298	11181	-0.13	-0.13	-0.12
Climax	61	11382	12084	12092	12053	-0.06	-0.06	-0.06
Hood	74	12884	12719	12724	13245	0.01	0.01	-0.03
Koon	110	16150	15549	15713	14995	0.04	0.03	0.07
Zuni	3500	24076	25195	27341	27282	-0.05	-0.14	-0.13
Tewa	5000	30171	26613	29399	29520	0.12	0.03	0.02
Bravo	15000	34745	35450	37085	36118	-0.02	-0.07	-0.04
					FMD	0.08	0.06	0.01
					FRMS	0.2	0.19	0.18

Table A-4: Cloud Base Comparison of Models to Observation (rel to burst point)

Shot	Yield (kT)	Observed Cloud Base (m)	Calculated Cloud Top (m)			Fractional Deviation		
			1979	Corrected	Improved	1979	Corrected	Improved
Humboldt	0.0078	593	450	471	688	0.24	0.21	-0.16
Catron	0.021	277	646	724	957	-1.33	-1.61	-2.45
Vesta	0.024		1577	1601	1642			
DonaAna	0.037	568	1657	1682	1942	-1.92	-1.96	-2.42
Hidalgo	0.077	1048	1297	1381	1740	-0.24	-0.32	-0.66
Quay	0.079	960	921	923	1219	0.04	0.04	-0.27
Eddy	0.083	858	1477	1419	1634	-0.72	-0.65	-0.90
RioArriba	0.09	2108	1187	1238	1699	0.44	0.41	0.19
Wrangell	0.115	739	984	994	1289	-0.33	-0.35	-0.75
Franklin	0.14	2949	2674	2804	2984	0.09	0.05	-0.01
Wheeler	0.197	2825	2104	2114	2539	0.26	0.25	0.10
Ray	0.2	1089	1326	1343	1676	-0.22	-0.23	-0.54
Ruth Johnnie Boy	0.2	1949	1551	1705	2147	0.20	0.13	-0.10
	0.5	2240	1629	1638	1921	0.27	0.27	0.14
Laplace	1	2763	3054	3077	3417	-0.11	-0.11	-0.24
SantaFe	1.3	2229	2230	2226	2899	0.00	0.00	-0.30
Lea	1.4	1925	2468	2468	3054	-0.28	-0.28	-0.59
Mora	2	1315	2845	2871	3150	-1.06	-1.18	-1.40
John	2		2702	2708	3100			

Table A-5: Cloud Base Comparison of Models to Observation (rel to burst point)

Shot	Yield (kT)	Observed Cloud Base (m)	Calculated Cloud Top (m)			Fractional Deviation		
			1979	Corrected	Improved	1979	Corrected	Improved
DeBaca	2.2	1315	2372	2358	3367	-0.80	-0.79	-1.56
Franklin Prime	4.7	4896	3601	3605	3979	0.26	0.26	0.19
Sanford	4.9	2415	3112	3129	3827	-0.29	-0.30	-0.58
Socorro	6	4378	3825	3839	4292	0.13	0.12	0.02
Morgan	8	6488	4183	4187	4594	0.36	0.35	0.29
Owens	9.7	4659	5464	5442	5107	-0.17	-0.17	-0.10
Wilson	10	6178	4209	4193	5169	0.32	0.32	0.16
Kepler	10	4630	5100	5077	5172	-0.10	-0.10	-0.12
Fizeau	11	6849	5241	5268	5451	0.23	0.23	0.20
Galileo	11	3734	5017	5021	5471	-0.34	-0.34	-0.47
Doppler	11	5264	5045	5027	4940	0.04	0.05	0.06
Dixie	11	6996	5260	5251	5713	0.25	0.25	0.18
Boltzman	12	5567	7258	7246	6759	-0.30	-0.30	-0.21
Newton	12	4058	5181	5218	5411	-0.28	-0.29	-0.33
Charleston	12	4354	4643	4684	4806	-0.07	-0.08	-0.10
Grable	15	5913	3866	3861	4233	0.35	0.35	0.28
Annie	16	7216	5917	5953	6614	0.18	0.18	0.08
Shasta	17	3387	6118	6036	5872	-0.81	-0.78	-0.73

Table A-6: Cloud Base Comparison of Models to Observation (rel to burst point)

Shot	Yield (kT)	Observed Cloud Base (m)	Calculated Cloud Top (m)			Fractional Deviation		
			1979	Corrected	Improved	1979	Corrected	Improved
Diablo	17	4581	5960	5920	6306	-0.30	-0.29	-0.38
Whitney	19	3967	5570	5568	6133	-0.40	-0.40	-0.55
Stokes	19	6497	5449	5439	5758	0.16	0.16	0.11
Badger	23	5550	4758	4772	5762	0.14	0.14	-0.44
Nancy	24	6520	5782	5801	6214	0.11	0.11	0.05
Encore	27	6858	5474	5476	5981	0.20	0.20	0.13
Harry	32	7070	7213	7377	8105	-0.02	-0.04	-0.15
Priscilla	37	6164	6914	6909	7392	-0.12	-0.12	-0.20
Lacrosse	40	6096	5180	5856	6011	0.15	0.04	0.01
Simon	43	8065	7907	7948	8090	0.02	0.01	0.00
Smoky	44		7566	7566	7677			
Climax	61	9035	7883	7891	8141	0.13	0.13	0.10
Hood	74	8921	8065	8065	8737	0.10	0.10	0.02
Koon	110		9875	10035	9649			
Zuni	3500	14932	13623	15016	15374	0.09	-0.01	-0.03
Tewa	5000		14017	16369	16435			
Bravo	15000	16853	20938	22055	20240	-0.24	-0.31	-0.20
					FMD	-0.12	-0.14	-0.29
					FRMS	0.47	0.49	0.66

Appendix B: Model Code

```
%*****
%*****
% Program: Thermal Bubble Modeling Program
% Programmer:  Capt Karson A. Sandman
% Date Programmed:  13 Jan 05
% Revision 0.43
%*****
%*****
% Description:  This program's intended use is for a two dimensional
% analysis of the rise of a thermal perturbation (bubble) that develops
% in the atmosphere such as that from the explosion of a nuclear
% detonation.  The model accounts for atmospheric moisture content and
% the shear effects that are a result of the varying horizontal
% atmospheric winds.
%*****
%*****

%*****
% Initially clear the workspace
clc
clear all
close all

%*****
% variable declarations

epsilon=.622; %[unitless]; ratio of dry air gas constant/vapor gas
% constant
lv=2.5e6; %[J/kg]; latent heat of vapor for condensation
Mv=.018015; %[kg/kmol]; molecular weight of water
R_star=8314.3; %[J/(kmol K)]; Universal gas constant
R_v=461.50; %[J/(kg*K)]; gas constant for water vapor
R_d=287.04; %[J/(kg*K)]; gas constant for dry air
M_avg=28.94; %[kg/kmol]; p76 Bohren & Albrecht; weight of air up to 100
% km altitude
Ts=373.16; %[K]; constant for Goff-Gratch Equation (METG 510 Class
%Notes)
Nz=180; %number of vertical elements
Nx=180; %number of horizontal elements
theta_bar(1:Nz,1)=0.0; %[K]; average potential temperature for each
% sounding level
qv_bar(1:Nz,1)=0.0; %[g/g]; average specific humidity for each sounding
% level
qv_max=.014; %[g/g]
Dz=100; %[m]; vertical grid spacing
Dx=100; %[m]; horizontal grid spacing
Cp=1004.0; %[ ]; constant pressure specific heat
Rgas=287.04; %[ ]; dry air gas constant
g=9.806; %[m/s^2]; acceleration due to gravity
p_surf=100000.0; %[Pa]; ref sea level pressure
pi_bar(1:Nz,1)=0.0; %Exner Function value
p_bar(1:Nz,1)=0.0; %[Pa]; average atmospheric pressure for each
% elevation
```

```

u_bar(1:Nz,1)=0.0; %[m/s]; average wind speed for each elevation
e_vap_press(1:Nz,1)=0.0; %[Pa]; average partial vapor pressure for each
%elevation
e_vap_press_s(1:Nz,1)=0.0; %[Pa]; average saturated partial vapor
%pressure for each elevation
rho_bar(1:Nz,1)=0.0; %[kg/m^3]; average density of atmosphere at each
%elevation

load tlh2.txt %loads the specified weather sounding; for this program,
%the weather should be gridded for 100m x 100m analysis points; in this
%case, the weather sounding was taken from Tallahassee, Fl, 1200Z on 14
%Jul, 2003; conversions performed as necessary

p_bar=tlh2(1:Nz,2)*10.0; %[Pa]
temp=(tlh2(1:Nz,4))/10+273.15; %[K]
temp_dewpt=(tlh2(1:Nz,5))/10+273.15; %[K]
wind_direction=(tlh2(1:Nz,6)); %[degrees]; direction 'from'
wind_speed=(tlh2(1:Nz,7))*0.5144444; %[m/s]; converted from knots

%determine the physical atmospheric properties from the sounding
for i=1:Nz

    e_vap_press(i)=10.0^(-7.90298*(Ts/temp_dewpt(i)-1)+ 5.02808...
        *log10(Ts/temp_dewpt(i)-1.3816E-7*(10^(11.344*(1-...
            temp_dewpt(i)/Ts))-1)+8.1328E-3*(10^(-3.49149...
            *(Ts/temp_dewpt(i)-1)) -1)+log10(101324.6))); %[Pa]; Goff-
            % Gratch equation converted from [Pa]
    e_vap_press_s(i)=10.0^(-7.90298*(Ts/temp(i)-1)+5.02808...
        * log10(Ts/temp(i)-1.3816E-7*(10^(11.344*(1-temp(i)...
            /Ts))-1)+8.1328E-3*(10^(-3.49149*(Ts/temp(i)-1))- ...
            1)+log10(101324.6))); %[Pa]; Goff-Gratch equation converted
            % from [Pa]
    theta_bar(i)=temp(i)*(p_surf/p_bar(i))^(Rgas/Cp); %[K]
    rho_bar(i)=p_bar(i)/(R_d*temp(i))*(1-(e_vap_press(i)/ ...
        p_bar(i))*(1-epsilon)); %[kg/m^3]
    qv_bar(i)=epsilon*e_vap_press(i)/(p_bar(i)-(1.0-...
        epsilon)*e_vap_press(i)); %[unitless]
    pi_bar(i)=(p_bar(i)/p_surf)^(Rgas/Cp);

end

%*****
% wind speed: takes only the east-west component for the horizontal
% wind; easily expanded to determine horizontal winds at another
% rotation

for i=1:Nz %develops the x component of the horizontal wind (wind is
%from the direction listed in sounding)

    if (wind_direction(i)>=0.0 & wind_direction(i)<90.0)
        u_bar(i)=-(sin(wind_direction(i)*pi/180.0)*wind_speed(i));
    elseif(wind_direction(i)>=90.0 & wind_direction(i)<180.0)
        u_bar(i)=-(sin(wind_direction(i)*pi/180.0)*wind_speed(i));
    elseif (wind_direction(i)>=180.0 & wind_direction(i)<270.0)
        u_bar(i)=-(sin(wind_direction(i)*pi/180.0)*wind_speed(i));

```

```

elseif (wind_direction(i)>=270.0 & wind_direction(i)<360.0)
    u_bar(i)=-(sin(wind_direction(i)*pi/180.0)*wind_speed(i));
end

end

%initially set the 2-d winds to the average at that elevation; the
%perturbation will then effect additional change

for i=1:Nx
    for j=1:Nz
        u(i,j)=u_bar(j);
        qv(i,j)=qv_bar(j);
    end
end

clear t1h2;
clear wind_direction;
clear wind_speed;

x(1:Nx,1)=0.0;
z(1:Nz,1)=0.0;

%set the horizontal and vertical axes to km for bookkeeping
for i=1:Nx
    for j=1:Nz
        x(i)=i*Dx/1000.0; %[km]
        z(j)=j*Dz/1000.0; %[km]
    end
end

% Display the horizontal wind profile
figure(1)
plot(u_bar,z);
xlabel('Horizontal Wind [m/s]');
ylabel('Height [km]');
%title('Component u');

% Display the vertical pressure profile
figure(2)
plot(p_bar,z);
xlabel('Average Pressure [Pa]');
ylabel('Height [km]');

% Display the vertical potential temperature profile
figure(3)
plot(theta_bar,z);
xlabel('Potential Temp [K]');
ylabel('Height [km]');

%*****
****
%initialize thermal bubble using potential temperature

```

```

del_theta=input('Enter the differential potential temp: [K]\n');
%bubble difference from surrounding
theta_prime(1:Nx,1:Nz)=0.0;
d_pi_prime_dz(1:Nx,1:Nz)=0.0;
theta(1:Nx,1:Nz)=0.0;
p(1:Nx,1:Nz)=0.0;
p_prime(1:Nx,1:Nz)=0.0;
pi_prime(1:Nx,1:Nz)=0.0;

%*****
****
% location and radius of the initial thermal bubble
x_not=9000; %[m]
z_not=1500; %[m]
r_not=input('Enter the initial perturbation bubble size [m]: \n');

%assign perturbation potential temp to the thermal bubble
for i=1:Nx
    for j=1:Nz
        r=min(((Dx*i-x_not)^2+(Dz*j-z_not)^2)^(1.0/2.0),r_not);
        theta_prime(i,j)=del_theta*.5*(cos(pi*r/r_not)+1);
        if (theta_prime(i,j)~=0.0) % this makes it a uniform temperature
                                %as per Bridgman
            theta_prime(i,j)=del_theta;
        end
        theta(i,j)=theta_prime(i,j)+theta_bar(j);
    end
end

%calculate the change in the Exner Function with elevation for the
%perturbation
%for i=1:Nx
%    % sum=0.0;
%    % for j=Nz-1:-1:1
%        % d_pi_prime_dz(i,j)=(g/(Cp*theta_bar(j))*...
%        (theta_prime(i,j))/theta_bar(j));
%    %end
%end

%Integration of the Exner Function
%for i=1:Nx
%    %for j=Nz-1:-1:1
%        %pi_prime(i,j)=pi_prime(i,j+1)+ d_pi_prime_dz(i,j)*Dz;
%        % 2d exner function
%    %end
%end

%determine the perturbation pressure and overall pressure from the
%Exner Function
for i=1:Nx
    for j=1:Nz
        %p_prime(i,j)=100000.0*(pi_prime(i,j)+...
        pi_bar(j))^(Cp/Rgas)-p_bar(j);%perturbation pressure 2d
        %if (abs(p_prime(i,j))<.001)
        %p_prime(i,j)=0.0;

```



```

        %end
        p(i,j)=p_prime(i,j)+p_bar(j);
    end
end

%compare_p=max(max(p_prime));

clear pi_prime;
clear d_pi_prime_dz;

%set the horizontal and vertical axes to km for bookkeeping
for i=1:Nx
    for j=1:Nz
        x(i)=i*Dx/1000.0; %[km]
        z(j)=j*Dz/1000.0; %[km]
    end
end

%*****
%*****
%Create Equation of motion arrays for horizontal and vertical winds,
pressure
%potential temp, specific humidity, saturation specific humidity

%variables below in box can be readily adjusted to maximize
effectiveness
%of the model
%*****
%
%          knob variables          *
del_time=input('Enter the desired time step [s],\n'); %
run_time=input('Enter the desired analysis time [s],\n'); %[s]*
N=4.0; %
gamma=.09;% filtering coefficient
del_ts=del_time/N; %time step for the Leapfrog integrations
cs=100.0; %[m/s]; rate of speed for the escaping gravity waves *
K=100.0; %[m^2/s]; diffusion coefficient
n=run_time/(2.0*del_time); % sets the appropriate number of steps*
%to run the model; for larger perturbations (high potential
%temp) the run time should be several hundred seconds
%*****

%declare and zero the working matrices
u_old(1:Nx,1:Nz)=0.0;
u_new(1:Nx,1:Nz)=0.0;
fu(1:Nx,1:Nz)=0.0;

w(1:Nx,1:Nz)=0.0;
w_old(1:Nx,1:Nz)=0.0;
w_new(1:Nx,1:Nz)=0.0;
fw(1:Nx,1:Nz)=0.0;

p_old(1:Nx,1:Nz)=0.0;
p_new(1:Nx,1:Nz)=0.0;
fp(1:Nx,1:Nz)=0.0;

```

```

theta_old(1:Nx,1:Nz)=0.0;
theta_new(1:Nx,1:Nz)=0.0;
ft(1:Nx,1:Nz)=0.0;

q_vs(1:Nx,1:Nz)=0.0;
qv_old(1:Nx,1:Nz)=qv;
qv_new(1:Nx,1:Nz)=qv;
qv_old_star(1:Nx,1:Nz)=0.0;
qv_prime(1:Nx,1:Nz)=0.0;
fqv(1:Nx,1:Nz)=0.0;

qc(1:Nx,1:Nz)=0.0;
qc_old(1:Nx,1:Nz)=0.0;
qc_new(1:Nx,1:Nz)=0.0;
qc_old_star(1:Nx,1:Nz)=0.0;
qc_prime(1:Nx,1:Nz)=0.0;
fqc(1:Nx,1:Nz)=0.0;

prod(1:Nx,1:Nz)=0.0;

rho_bar_star=rho_bar*0.0;

%splits the vertical value for the density for the grid point system
for j=1:Nz-1
    rho_bar_star(j)=(rho_bar(j+1)+rho_bar(j))/2.0;
end
rho_bar_star(Nz)=rho_bar(Nz);

%initially set all of the arrays equal to each other
u_old=u;
u_new=u;

p_old=p;
p_new=p;

theta_old=theta;
theta_new=theta;

%initiate a movie file to track the changes in the system
aviobj1 = avifile('mymovie1.avi','fps',10);

%*****
*
%*****
*

%initiate the numerical integration scheme to track the weather changes

%Large Step Routine (which contains the smallstep routine)
for k=1:n

    qv_old_star=qv_prime;
    qc_old_star=qc_prime;
    theta_old_star=theta_prime;

```

```

for i=1:Nx
    for j=1:Nz
        u_old_star(i,j)=u_old(i,j)-u_bar(j);
    end
end

%integration scheme for potential temperature
for i=1:Nx
    for j=1:Nz
        if ((i==1|i==Nx)&(j~=1)&(j~=Nz))
            ft(i,j)=0.0;
        elseif ((j==1|j==Nz)&(i~=1)&(i~=Nz))
            ft(i,j)=0.0;
        elseif ((i==1&j==1)|(i==1&j==Nz)|(i==Nx&j==1)|(i==Nx&j==Nz))
            ft(i,j)=0.0;
        else
            ft(i,j)=1.35*K*((theta_old(i+1,j)- ...
                2.0*theta_old(i,j)+theta_old(i-1,j))/Dx^2+...
                (theta_old_star(i,j+1)-2.0*theta_old_star(i,j)+ ...
                theta_old_star(i,j-1))/Dz^2)...
                -1.0/2.0*(u(i+1,j)*(theta(i+1,j)-theta(i,j))/Dx+...
                u(i,j)*(theta(i,j)-theta(i-1,j))/Dx)-...
                1.0/2.0*(w(i,j+1)*(theta(i,j+1)-theta(i,j))/Dz+...
                w(i,j)*(theta(i,j)-theta(i,j-1))/Dz));
            end
        end
    end
end

%Smallstep Routine: does the integration for the slow processes using a
%smaller time step to prevent anomolies in the integration (this is the
%Leapfrog portion)
for m=0:(2*N-1)

    %integration scheme for the specific humidity
    for i=1:Nx
        for j=1:Nz
            if ((i==1|i==Nx)&(j~=1)&(j~=Nz))
                fqv(i,j)=0.0;
            elseif ((j==1|j==Nz)&(i~=1)&(i~=Nz))
                fqv(i,j)=0.0;
            elseif ((i==1&j==1)|(i==1&j==Nz)|(i==Nx&j==1)|(i==Nx&j==Nz))
                fqv(i,j)=0.0;
            else
                fqv(i,j)=K*((qv_old(i+1,j)-2.0*qv_old(i,j)
                    +qv_old(i-1,j))/Dx^2 ...
                    +(qv_old_star(i,j+1)-2.0*qv_old_star(i,j)+ ...
                    qv_old_star(i,j-1))/Dz^2)...
                    -1.0/2.0*((qv(i+1,j)+qv(i,j))/2.0*...
                    (qv(i+1,j)-qv(i,j))/Dx...
                    +(qv(i,j)+qv(i-1,j))/2.0*(qv(i,j)-qv(i-1,j))/Dx)...
                    -1.0/2.0*((w(i,j+1)+w(i-1,j+1))/2.0*(qv(i,j+1)- ...
                    qv(i,j))/Dz +(w(i,j)+w(i-1,j))/2.0*...
                    (qv(i,j)-qv(i,j-1))/Dz));
            end
        end
    end
end

```

```

end
end

%integration scheme for saturation specific humidity
for i=1:Nx
    for j=1:Nz
        if ((i==1|i==Nx)&(j~=1)&(j~=Nz))
            fqc(i,j)=0.0;
        elseif ((j==1|j==Nz)&(i~=1)&(i~=Nz))
            fqc(i,j)=0.0;
        elseif ((i==1&j==1)|(i==1&j==Nz)|(i==Nx&j==1)|(i==Nx&j==Nz))
            fqc(i,j)=0.0;
        else
            fqc(i,j)=K*((qc_old(i+1,j)-2.0*qc_old(i,j)+qc_old(i-...
                1,j))/Dx^2 +(qc_old_star(i,j+1)- ...
                2.0*qc_old_star(i,j)+ qc_old_star(i,j-1))/Dz^2)...
                -1.0/2.0*((qc(i+1,j)+qc(i,j))/2.0*(qc(i+1,j)- ...
                qc(i,j))/Dx +(qc(i,j)+qc(i-1,j))/2.0*(qc(i,j) ...
                -qc(i-1,j))/Dx) -1.0/2.0*((w(i,j+1)+ ...
                w(i-1,j+1))/2.0*(qc(i,j+1)-qc(i,j))/Dz...
                +(w(i,j)+w(i-1,j))/2.0*(qc(i,j)-qc(i,j-1))/Dz));
        end
    end
end
end

for i=1:Nx
    for j=1:Nz
        qv_new(i,j)=qv_old(i,j)+del_ts*fqv(i,j);
        qc_new(i,j)=qc_old(i,j)+del_ts*fqc(i,j);
        if (qc_new(i,j)<0.0)
            qc_new(i,j)=0.0;
        end
    end
end

%integration scheme for the horizontal wind
for i=1:Nx
    for j=1:Nz
        if ((i==1|i==Nx)&(j~=1)&(j~=Nz))
            fu(i,j)=0.0;
        elseif ((j==1|j==Nz)&(i~=1)&(i~=Nz))
            fu(i,j)=0.0;
        elseif
            ((i==1&j==1)|(i==1&j==Nz)|(i==Nx&j==1)|(i==Nx&j==Nz))
            fu(i,j)=0.0;
        else
            fu(i,j)=K*((u_old(i+1,j)-2.0*u_old(i,j) ...
                +u_old(i-1,j))/Dx^2 +(u_old_star(i,j+1)- ...
                2.0*u_old_star(i,j)+u_old_star(i,j-1))/Dz^2)...
                -1.0/2.0*((u(i+1,j)+u(i,j))/2.0*(u(i+1,j)- ...
                u(i,j))/Dx +(u(i,j)+u(i-1,j))/2.0*(u(i,j)- ...
                u(i-1,j))/Dx)-1.0/2.0*((w(i,j+1)+ ...
                w(i-1,j+1))/2.0*(u(i,j+1)-u(i,j))/Dz...
                +(w(i,j)+w(i-1,j))/2.0*(u(i,j)-u(i,j-1))/Dz));
        end
    end
end

```

```

        end
    end

    %integration scheme for the vertical wind
    for i=1:Nx
        for j=1:Nz
            if ((i==1|i==Nx)&(j~=1)&(j~=Nz))
                fw(i,j)=0.0;
            elseif ((j==1|j==Nz)&(i~=1)&(i~=Nz))
                fw(i,j)=0.0;
            elseif
                ((i==1&j==1)|(i==1&j==Nz)|(i==Nx&j==1)|(i==Nx&j==Nz))
                fw(i,j)=0.0;
            else
                fw(i,j)=g/2.0*((theta(i,j)-theta_bar(j)) ...
                    /theta_bar(j)+(theta(i,j-1)-theta_bar(j-...
                    1))/theta_bar(j-1))+K*((w_old(i+1,j)- ...
                    2.0*w_old(i,j)+w_old(i-1,j))/Dx^2 ...
                    +(w_old(i,j+1)-2.0*w_old(i,j)+w_old(i,j-...
                    1))/Dz^2)- 1.0/2.0*((u(i+1,j)+u(i,j)) ...
                    /2.0*(w(i+1,j)-w(i,j))/Dx...
                    +(u(i,j)+u(i-1,j))/2.0*(w(i,j)-w(i-1,j))/Dx)...
                    -1.0/2.0*((w(i,j+1)+w(i,j))/2.0*(w(i,j+1)- ...
                    w(i,j))/Dz+(w(i,j)+w(i,j-1))/2.0*(w(i,j)-w(i,j-...
                    1))/Dz);
            end
        end
    end

    %incrementation step for the horizontal and vertical winds
    for i=1:Nx
        for j=1:Nz
            if ((i==1|i==Nx)&(j~=1)&(j~=Nz))
                u_new(i,j)=u_bar(j);
            elseif ((j==1|j==Nz)&(i~=1)&(i~=Nz))
                w_new(i,j)=0.0;
            elseif
                ((i==1&j==1)|(i==1&j==Nz)|(i==Nx&j==1)|(i==Nx&j==Nz))
                u_new(i,j)=u_bar(j);
                w_new(i,j)=0.0;
            else
                u_new(i,j)=u(i,j)-del_ts/(rho_bar(j)*Dx)* ...
                    (p_prime(i,j)-p_prime(i-1,j))+del_ts*fu(i,j);
                w_new(i,j)=w(i,j)-del_ts/(rho_bar_star(j)*Dz)* ...
                    (p_prime(i,j)-p_prime(i,j-1))+del_ts*fw(i,j);
            end
        end
    end

    %integration scheme for the pressure
    for i=1:Nx
        for j=1:Nz
            if ((i==1|i==Nx)&(j~=1)&(j~=Nz))
                p_new(i,j)=p_bar(j);
            end
        end
    end

```

```

elseif ((j==1|j==Nz)&(i~=1)&(i~=Nz))
    p_new(i,j)=p_old(i,j);
elseif
((i==1&j==1)|(i==1&j==Nz)|(i==Nx&j==1)|(i==Nx&j==Nz))
    p_new(i,j)=p_bar(j);
else
    p_new(i,j)=p(i,j)-cs^2*del_ts*(rho_bar(j)* ...
        (u_new(i+1,j)-u_new(i,j))/Dx...
        +(rho_bar_star(j+1)*w_new(i,j+1)- ...
        rho_bar_star(j)*w_new(i,j))/Dz);
    end
end
end

end %end of smallstep routine

%comparator step to track specific humidity in the system
for i=1:Nx
    for j=1:Nz
q_vs(i,j)=380.0/p_bar(j)*exp(17.27*(pi_bar(j)*theta_new(i,j)-273)...
        /(pi_bar(j)*theta_new(i,j)-36));
        prod(i,j)=(qv(i,j)-q_vs(i,j))*(1+(lv*4093*q_vs(i,j)) ...
        /(Cp*(pi_bar(j)*theta_new(i,j)-36)^2));
        prod(i,j)=max(prod(i,j),-qc(i,j));
    end
end

prod(i,j)=max(prod(i,j),-qc(i,j));

%advances the potential temperature with the overall time step
theta_new=theta_old + 2.0*del_time*ft;

%updates the potential temperature with specific humidity
for i=1:Nx
    for j=1:Nz
        theta_new(i,j)=theta_new(i,j)+lv*prod(i,j)/(Cp*pi_bar(j));
        qv_new(i,j)=max(qv_new(i,j)-prod(i,j),0);
        qc_new(i,j)=max(qc_new(i,j)-prod(i,j),0);
    end
end

end

%*****
%*****
%Asselin Filter

u=u+gamma*(u_old-2.0*u+u_new);
w=w+gamma*(w_old-2.0*w+w_new);
p=p+gamma*(p_old-2.0*p+p_new);
theta=theta+gamma*(theta_old-2.0*theta+theta_new);
qc=qc+gamma*(qc_old-2.0*qc+qc_new);
qv=qv+gamma*(qv_old-2.0*qv+qv_new);

u_old=u;

```

```

w_old=w;
p_old=p;
theta_old=theta;
qc_old=qc;
qv_old=qv;

u=u_new;
w=w_new;
p=p_new;
theta=theta_new;
qc=qc_new;
qv=qv_new;

%determine the new perturbation values after integrating and

for i=1:Nx
    for j=1:Nz
        theta_prime(i,j)=theta_old(i,j)-theta_bar(j);
        if (theta_prime(i,j)>del_theta)
            theta_prime(i,j)=del_theta;
        end

        theta_old(i,j)=theta_bar(j)+theta_prime(i,j);
        p_prime(i,j)=p_old(i,j)-p_bar(j);
        qc_prime(i,j)=qc_old(i,j);
        qv_prime(i,j)=qv_old(i,j)-qv_bar(j);
    end
end

time=2.0*del_time*k %bookkeeper for overall time

%develop the plot to track the changes in the weather model
if (time>=10 & time<10.121)

    dlmwrite('w_10_s.txt', w, 'precision', '%12.8f', 'newline',
'pc')%Need to put this data into an overall matrix and then save to
text file
    dlmwrite('u_10_s.txt', u, 'precision', '%12.8f', 'newline',
'pc')
    dlmwrite('theta_prime_10_s.txt', theta_prime, 'precision',
'%12.8f', 'newline', 'pc')
    dlmwrite('p_prime_10_s.txt', p_prime, 'precision', '%12.8f',
'newline', 'pc')

    figure(4)
    subplot(2,2,1); [C,h]=contourf(x,z,p_prime',10);
    colormap(jet);
    h = clabel(C,h,'manual');
    set(h,'BackgroundColor',[1 1 .47]);
    timetitle=10;
    title(['Pert. Press. [Pa]; t = ',int2str(timetitle),
'[s]'])

    xlabel('Domain Width, [km]');
    ylabel('Domain Height, [km]');
    grid on

```

```

subplot(2,2,2); [C,h]=contourf(x,z,theta_prime',8);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=10;
title(['Pert. Theta [K]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,3); [C,h]=contourf(x,z,w',8);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=10;
title(['w [m/s]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,4); [C,h]=contourf(x,z,u',10);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=10;
title(['u [m/s]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

    end %of the weather 10 s plot

if (time>=30 & time<30.121)

    dlmwrite('w_30_s.txt', w', 'precision', '%12.8f', 'newline',
'pc')%Need to put this data into an overall matrix and then save to
text file
    dlmwrite('u_30_s.txt', u', 'precision', '%12.8f', 'newline',
'pc')
    dlmwrite('theta_prime_30_s.txt', theta_prime', 'precision',
'%12.8f', 'newline', 'pc')
    dlmwrite('p_prime_30_s.txt', p_prime', 'precision', '%12.8f',
'newline', 'pc')

figure(6)
subplot(2,2,1); [C,h]=contourf(x,z,p_prime',10);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=30;
title(['Pert. Press. [Pa]; t = ',int2str(timetitle),
'[s]'])

xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');

```



```

grid on

subplot(2,2,2); [C,h]=contourf(x,z,theta_prime',8);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=30;
title(['Pert. Theta [K]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,3); [C,h]=contourf(x,z,w',8);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=30;
title(['w [m/s]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,4); [C,h]=contourf(x,z,u',10);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=30;
title(['u [m/s]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

end %of the weather 30 s plot

if (time>=60 & time<60.121)

    dlmwrite('w_60_s.txt', w, 'precision', '%12.8f', 'newline',
'pc')%Need to put this data into an overall matrix and then save to
text file
    dlmwrite('u_60_s.txt', u, 'precision', '%12.8f', 'newline',
'pc')
    dlmwrite('theta_prime_60_s.txt', theta_prime, 'precision',
'%12.8f', 'newline', 'pc')
    dlmwrite('p_prime_60_s.txt', p_prime, 'precision', '%12.8f',
'newline', 'pc')

    figure(7)
    subplot(2,2,1); [C,h]=contourf(x,z,p_prime',10);
    colormap(jet);
    h = clabel(C,h,'manual');
    set(h,'BackgroundColor',[1 1 .47]);
    timetitle=60;
    title(['Pert. Press. [Pa]; t = ',int2str(timetitle),
'[s]'])
    xlabel('Domain Width, [km]');

```

```

ylabel('Domain Height, [km]');
grid on

subplot(2,2,2); [C,h]=contourf(x,z,theta_prime',8);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=60;
title(['Pert. Theta [K]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,3); [C,h]=contourf(x,z,w',8);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=60;
title(['w [m/s]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,4); [C,h]=contourf(x,z,u',10);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=60;
title(['u [m/s]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

    end %of the weather 60 s plot

if (time>=180 & time<180.121)

    dlmwrite('w_180_s.txt', w, 'precision', '%12.8f', 'newline',
'pc')%Need to put this data into an overall matrix and then save to
text file
    dlmwrite('u_180_s.txt', u, 'precision', '%12.8f', 'newline',
'pc')
    dlmwrite('theta_prime_180_s.txt', theta_prime, 'precision',
'%12.8f', 'newline', 'pc')
    dlmwrite('p_prime_180_s.txt', p_prime, 'precision', '%12.8f',
'newline', 'pc')

    figure(9)
    subplot(2,2,1); [C,h]=contourf(x,z,p_prime',10);
    colormap(jet);
    h = clabel(C,h,'manual');
    set(h,'BackgroundColor',[1 1 .47]);
    timetitle=180;
    title(['Pert. Press. [Pa]; t = ',int2str(timetitle),
'[s]'])

```

```

xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,2); [C,h]=contourf(x,z,theta_prime',8);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=180;
title(['Pert. Theta [K]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,3); [C,h]=contourf(x,z,w',8);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=180;
title(['w [m/s]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,4); [C,h]=contourf(x,z,u',10);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=180;
title(['u [m/s]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

end %of the weather 180 s plot

if (time>=270 & time<270.121)

    dlmwrite('w_270_s.txt', w, 'precision', '%12.8f', 'newline',
'pc')%Need to put this data into an overall matrix and then save to
text file
    dlmwrite('u_270_s.txt', u, 'precision', '%12.8f', 'newline',
'pc')
    dlmwrite('theta_prime_270_s.txt', theta_prime, 'precision',
'%12.8f', 'newline', 'pc')
    dlmwrite('p_prime_270_s.txt', p_prime, 'precision', '%12.8f',
'newline', 'pc')

figure(11)
subplot(2,2,1); [C,h]=contourf(x,z,p_prime',10);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=270;

```

```

title(['Pert. Press. [Pa]; t = ',int2str(timetitle),
'[s]'))

xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,2); [C,h]=contourf(x,z,theta_prime',8);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=270;
title(['Pert. Theta [K]; t = ',int2str(timetitle), '[s]'))
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,3); [C,h]=contourf(x,z,w',8);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=270;
title(['w [m/s]; t = ',int2str(timetitle), '[s]'))
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,4); [C,h]=contourf(x,z,u',10);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);
timetitle=270;
title(['u [m/s]; t = ',int2str(timetitle), '[s]'))
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

end %of the weather 180 s plot

if (time>=350 & time<350.121)

    dlmwrite('w_350_s.txt', w, 'precision', '%12.8f', 'newline',
'pc')%Need to put this data into an overall matrix and then save to
text file
    dlmwrite('u_350_s.txt', u, 'precision', '%12.8f', 'newline',
'pc')
    dlmwrite('theta_prime_350_s.txt', theta_prime, 'precision',
'%12.8f', 'newline', 'pc')
    dlmwrite('p_prime_350_s.txt', p_prime, 'precision', '%12.8f',
'newline', 'pc')

figure(12)
subplot(2,2,1); [C,h]=contourf(x,z,p_prime',10);
colormap(jet);
h = clabel(C,h,'manual');
set(h,'BackgroundColor',[1 1 .47]);

```

```

        timetitle=350;
        title(['Pert. Press. [Pa]; t = ',int2str(timetitle),
'[s]'))

        xlabel('Domain Width, [km]');
        ylabel('Domain Height, [km]');
        grid on

        subplot(2,2,2); [C,h]=contourf(x,z,theta_prime',8);
        colormap(jet);
        h = clabel(C,h,'manual');
        set(h,'BackgroundColor',[1 1 .47]);
        timetitle=350;
        title(['Pert. Theta [K]; t = ',int2str(timetitle), '[s]'))
        xlabel('Domain Width, [km]');
        ylabel('Domain Height, [km]');
        grid on

        subplot(2,2,3); [C,h]=contourf(x,z,w',8);
        colormap(jet);
        h = clabel(C,h,'manual');
        set(h,'BackgroundColor',[1 1 .47]);
        timetitle=350;
        title(['w [m/s]; t = ',int2str(timetitle), '[s]'))
        xlabel('Domain Width, [km]');
        ylabel('Domain Height, [km]');
        grid on

        subplot(2,2,4); [C,h]=contourf(x,z,u',10);
        colormap(jet);
        h = clabel(C,h,'manual');
        set(h,'BackgroundColor',[1 1 .47]);
        timetitle=350;
        title(['u [m/s]; t = ',int2str(timetitle), '[s]'))
        xlabel('Domain Width, [km]');
        ylabel('Domain Height, [km]');
        grid on

        end %of the weather 180 s plot

    if (mod(k/3.0,2)==0.0) %to make the movie

        figure(13)
        subplot(2,2,1); [C,h]=contourf(x,z,p_prime',10);
        colormap(jet);

        clabel(C,h,'FontSize',9,'Color','k','Rotation',0,'LabelSpacing',100);
        timetitle=2.0*k*del_time;
        title(['Pert. Press. [Pa]; t = ',int2str(timetitle),
'[s]'))

        xlabel('Domain Width, [km]');
        ylabel('Domain Height, [km]');
        grid on

```

```

subplot(2,2,2); [C,h]=contourf(x,z,theta_prime',10);
colormap(jet);

clabel(C,h,'FontSize',9,'Color','k','Rotation',0,'LabelSpacing',100);
timetitle=2.0*k*del_time;
title(['Pert. Theta [K]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,3); [C,h]=contourf(x,z,w',10);
colormap(jet);

clabel(C,h,'FontSize',9,'Color','k','Rotation',0,'LabelSpacing',100);
timetitle=2.0*k*del_time;
title(['w [m/s]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

subplot(2,2,4); [C,h]=contourf(x,z,u',10);
colormap(jet);

clabel(C,h,'FontSize',9,'Color','k','Rotation',0,'LabelSpacing',175);
timetitle=2.0*k*del_time;
title(['u [m/s]; t = ',int2str(timetitle), '[s]'])
xlabel('Domain Width, [km]');
ylabel('Domain Height, [km]');
grid on

drawnow

aviobj1 = addframe(aviobj1,figure(13));           % Add Figure
to avi
    end %of the movie plot

end %of the overall numerical integration loop
aviobj1 = close(aviobj1);
%end of the program

```

Appendix C: METG 612 2D Cloud Modeling Project

During the rest of the quarter, we are going to systematically build a 2D cloud model.

The idea is that we can add simple pieces to the model week by week until we have a full blown 2D moist cloud model by the end of the quarter. Each week you will hand in some verification that you have gotten the new pieces to work. Here is a schedule for this project. The dates are given for completion of these pieces of the model.

- 10 July: Set up the basic framework, create the base state arrays
- 17 July: 2D initialization
- 24 July: Integration of simplified momentum and pressure equations
- 31 July: Add theta equation, complete the u and w equations – Dry model complete!
- 7 August: Add in equations for q_v , q_c , and q_r . Add in microphysics – Moist model complete!

Basic reference paper: “Simulation of the thunderstorm sub-cloud environment” by Anderson et al. (Handout). This paper describes the basic dry problem we are going to try to make work and the quasi-compressible equations we are going to use.

Programming environment: IDL

Why? IDL is a great scientific language for both data analysis and display. Memory allocation in IDL is dynamic (like in C). This has lots of advantages. IDL also has array syntax statements, like $A(2:nx-1)=B(1:nx-2)*C(3:nx)$. This is a really fast way of programming and is VERY similar to the FORTRAN 90 standard. The graphics in IDL are very simple to use. If you use FORTRAN, you must use something like NCAR graphics to display the results which requires a lot more time to get the graphics right.

METG 612 2D Cloud Modeling Project

Assignment 1 – Due 10 July 2000

It is going to be very important that we get started in a certain way. You need to be thinking of creating a program with 5-10 modular pieces (subroutines) in it. In this assignment, you will build the first piece of it, the initialization of 1D arrays which will describe the base state variables for the model equations in this first assignment.

I think that good code design is very similar to a good outline for a paper. For example:

PRO Main

- Specify model constants

- Specify domain and integration characteristics

- *Initialize 1D arrays (subroutine)

- *Initialize 2D arrays (subroutine)

- *Output max/min of 2D fields (output subroutine)

- *Plot 2D fields (plotting subroutine)

- Time step loop

 - Call solver

 - Integrate Theta Equation

 - *Compute advective terms for Theta

 - *Compute diffusive terms for Theta

 - *Update Theta

 - Compute RHS of U equation

 - *Compute advection for U


```

        *Compute diffusion for U
    Compute RHS of W equation
        *Compute advection for W
        *Compute diffusion for W
        *Compute buoyancy for W
    Update U and W
    Update Pressure equation
End solver
If it is time: Plot 2D fields
If it is time: Output max/min of 2D fields
End time step loop
END

```

This may be one way to think about designing the model. All of the ‘*’ quantities are additional procedures you have to write. This allows you to create pieces of code which can be changed in isolation and debugged by themselves.

The specific assignment: Set up the 1D initialization routines

Domain: 10 km wide (x), 5 km deep (z)

Grid: We will have 50 x 25 grid zones (scalar points), i.e., $Dx = Dz = 200$ m. ($NX = 50$, $NZ = 25$)

Constants: Besides the grid parameters, we need the following constants set up (more will be added later: $c_s = 75.0$ ($m\ s^{-1}$), $K = 100$ ($m^2\ s^{-1}$), $C_p = 1004.0$, $R_{gas} = 287.04$, $g = 9.806$ ($m\ s^{-2}$), $p_{sfc} = 1.0 \times 10^5$ (Pa))

Create a procedure which (1) initializes the grid information and constants, and (2) creates and returns all the base state variables.

Create the following arrays: sgx, sgz, tbar, qvbar, ubar, psfc

Dimension all the “bar” arrays to be NZ points

Dimension the sgx array to be NX, dimension the sgz array to be NZ. These arrays describe the position of the scalar grid points.

Pass to the procedure INIT1D the following information: domain size, # points in x, z direction.

Pass in the following arrays to the INIT1D: sgx, sgz, tbar, qvbar, ubar, psfc

Inside INIT1D set up code to create 2 soundings

Sounding 1 – Dry adiabatic sounding, no moisture or wind

$$tbar(z) = 300.0$$

$$qvbar(z) = 0.0$$

$$ubar(z) = 0.0$$

Generate pressure and density arrays using the hydrostatic equation and equation of states (EOS).

$$\frac{d\bar{\pi}}{dz} = \frac{-g}{c_p \bar{\theta}(z)} ; \pi_1 = \pi_{sfc} - \frac{g(\Delta z/2)}{c_p \bar{\theta}_1} ; \pi_k = \pi_{k-1} - \frac{g\Delta z}{c_p \left(\frac{\theta_k + \theta_{k-1}}{2} \right)}$$

Note that π is the Exner function given by $\pi = \left(\frac{p}{p_o} \right)^\kappa$.

Get $\bar{p}(z)$ from $\bar{\pi}(z)$, $\bar{\rho}(z)$ from Equations of State (EOS)

Sounding 2: Moist sounding

Weisman and Klemp, 1982, JAS, pp. 506.

$$\bar{\theta}(z) = \begin{cases} \theta_o + (\theta_{tr} - \theta_o) \left(\frac{z}{z_{tr}} \right)^{5/4} & z \leq z_{tr} \\ \theta_{tr} \exp \left[\frac{g}{c_p T_{tr}} (z - z_{tr}) \right] & z > z_{tr} \end{cases}$$

where

$$p_{tr} = 200mb = 2 \times 10^4 Pa, \theta_o = 300K, \theta_{tr} = 343K, T_{tr} = 213K, \text{ and } z_{tr} = 12km$$

for $\bar{q}_v(z)$ - first create humidity variable

$$H(z) = \begin{cases} 1 - \frac{3}{4} \left(\frac{z}{z_{tr}} \right)^{5/4} & z \leq z_{tr} \\ 0.25 & z > z_{tr} \end{cases}$$

then create $\bar{q}_v^*(z)$ via Teton's formula

$$\bar{q}_v^*(z) = \frac{380Pa}{\bar{p}(z)} \exp \left[17.27 \frac{(\bar{\pi}(z)\bar{\theta}(z) - 273K)}{(\bar{\pi}(z)\bar{\theta}(z) - 36K)} \right] \cdot H(z)$$

then

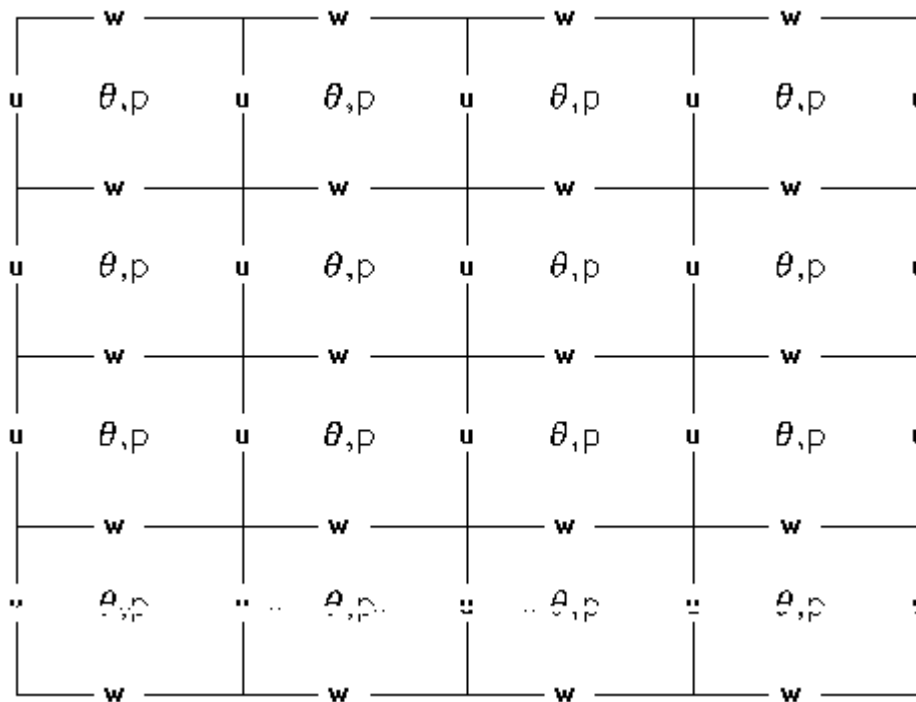
$$\bar{q}_v(z) = \min(q_{v_{\max}}, \bar{q}_v^*(z)), \text{ where } q_{v_{\max}} = 0.014 \frac{g}{g}$$

METG 612 2D Cloud Modeling Project

Assignment #2 – Due 17 July 2000

Today we create the initial state of the model. The initial condition (IC) will be similar to the Anderson et al. paper I handed out to you last week. A cold bubble will be placed in the domain with no flow. You will also create a plotting routine which you can call and will plot out all four of the dependent variables of your domain. Hand in plots of your initial conditions to me next week.

BASE STATE: Use sounding #1 (dry sounding)



Create the 2D arrays for u, w, t, and p, noting that because of the grid staggering, u & w will have extra points in the x and z directions, respectively. The grid will look like this, in abbreviated form:

Using a sub-procedure (INIT2D??):

Initialize the 2D variables using the base state information, note that “p” is the perturbation pressure, therefore DO NOT initialize it using the base state pressure.

Create a cold bubble in the theta field using the following equations and parameters

$$\theta(x, z) = \bar{\theta}(z) + \Delta\theta \cdot 0.5 \cdot \left(\cos\left(\frac{\pi \cdot r}{r_o}\right) + 1 \right) \quad (\text{Note: For this line, } \pi=3.1415\dots)$$

$$\text{where } r = \min\left[\sqrt{(x - x_o)^2 + (z - z_o)^2}, r_o\right]$$

$$\Delta\theta = -10K, \quad x_o = 5000m, \quad z_o = 2500m, \quad \text{and } r_o = 2000m$$

Now, initialize the (perturbation) pressure field using the hydrostatic equation. There are several ways to do this, but the easiest is to integrate DOWNWARD the following equation

$$\frac{\partial \pi'}{\partial z} = \frac{g}{c_p \bar{\theta}(z)} \left(\frac{\theta - \bar{\theta}(z)}{\bar{\theta}(z)} \right) \quad \text{where } \pi' = 0 \text{ at the top}$$

$$\text{and } p'(x, z) = 1.000 \times 10^5 \left(\pi'(x, z) + \bar{\pi}(z) \right)^{\frac{c_p}{R}} - \bar{p}(z)$$

Create a graphics program which will plot out a 4 panel display of p', θ', u, and w for this initial state. Make sure the time of the plot (in this case, t=0) is included in your labeling.

Note: u and w will be zero for this week's assignment. Those panels may be included, but not contoured. (i.e. Leave a space for each.)

IDL commands you might need include the following,

$y=(y < y_0)$

contour

!p.multi = [0,2,2]

xyouts

set_plot

device

METG 612 2D Cloud Modeling Project

Assignment #3 – Due 24 July 2000

Today we start creating motions in the model. We are NOT going to move the bubble yet, we are going to work the part of the equations which feed the negative buoyancy into the momentum equations and through the divergence, therefore affecting the pressure. Hand in plots of your results to me next week.

Equations of motion for Part 3:

$$\frac{\partial u}{\partial t} = -\frac{1}{\bar{\rho}} \frac{\partial p}{\partial x} \quad \frac{\partial w}{\partial t} = -\frac{1}{\bar{\rho}} \frac{\partial p}{\partial z} + g \frac{\theta - \bar{\theta}}{\bar{\theta}}$$

$$\frac{\partial p}{\partial t} = -c_s^2 \left(\bar{\rho} \frac{\partial u}{\partial x} + \frac{\partial(\bar{\rho} w)}{\partial z} \right) \quad \frac{\partial \theta}{\partial t} = 0$$

For simplicity, create four arrays for u and w (uold, u, unew, and fu, etc.). Later on I might show you how to eliminate one of those arrays. We are going to combine several different types of numerical methods for this model. First, the advection in the model will be done using a Leap-Frog time scheme. Therefore, on each time step, we will be integrating from old to new time levels. In the pressure gradient and divergence terms (this week's stuff), we will use the forward-backward scheme, which can be combined with the Leap-Frog scheme. At the end of the calculation, we need to do a time filter, called the Asselin filter so the Leap-Frog scheme does not separate onto two separate solutions on the space-time grid. So here is the procedure:

Create a routine called "Solver" in the main program. You call "Solver" once per large time step. Create a time stepping loop in the main program which will call "Solver" the correct number of times in order to integrate the solution from zero to time "t," in this

case, “t” = 300 sec. Since we choose a time step of 2 seconds, therefore the “Solver” procedure will be called 150 times. Inside this loop one can also put in the conditional statements which control plotting, etc.

Initialize all tendency arrays to zero. Compute the buoyancy term using the theta data.

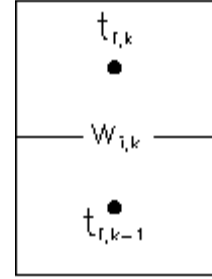
Perform the “small” step integration, integrating 2N steps from uold → unew, etc.

Apply the Asselin time filter.

Run the model for 300 seconds.

I will now outline these steps in some detail:

Later on, the fu and fw arrays will contain the advection and diffusion of u and w, the so-called “slow” processes. For now, set fu=0. The only “slow” process that will be included for now is the buoyancy term on the RHS of the vertical equation of motion. We are on a staggered grid, so that we need to vertically average the buoyancy to get it at the w-point.



$$fw_{i,k} = \frac{g}{2} \left(\frac{\theta_{i,k} - \bar{\theta}_k}{\bar{\theta}_k} + \frac{\theta_{i,k-1} - \bar{\theta}_{k-1}}{\bar{\theta}_{k-1}} \right)$$

Create the “smalstep” routine. Create a callable procedure which passes in all of the arrays and constants needed to integrate the equations listed below in the following manner:

Forward Step:

$$u_{i,k}^{m+1} = u_{i,k}^m - \frac{\Delta t_s}{\bar{\rho}_k \Delta x} (p_{i,k}^m - p_{i-1,k}^m) + \Delta t_s fw_{i,k}$$

$$w_{i,k}^{m+1} = w_{i,k}^m - \frac{\Delta t_s}{\overline{\rho}_k^* \Delta z} (p_{i,k}^m - p_{i,k-1}^m) + \Delta t_s f w_{i,k}$$

Backward Step:

$$p_{i,k}^{m+1} = p_{i,k}^m - c_s^2 \Delta t_s \left[\frac{\overline{\rho}_k (u_{i+1,k}^{m+1} - u_{i,k}^{m+1})}{\Delta x} + \frac{(\overline{\rho}_{k+1}^* w_{i,k+1}^{m+1} - \overline{\rho}_k^* w_{i,k}^{m+1})}{\Delta z} \right]$$

$$\text{where } \overline{\rho}_k^* = \frac{(\overline{\rho}_k + \overline{\rho}_{k-1})}{2}$$

Use the following boundary conditions, that w=0 at the top and bottom and that u=0 on the left and right boundaries. Inside the “Solver” routine, set up a loop which looks like:

$$\Delta t_s = \Delta t / N$$

Copy the “old” data arrays into the “new” data arrays (i.e. unew=uold, etc. which is similar to saying $u^{n+1} = u^{n-1}$).

FOR m = 0,2*N-1 do begin

SMLSTEP(unew, fu, wnew, fw, pnew, delta_ts, rho_bar, ...)

ENDFOR

This will then do the “time-splitting” for you. At the end of this loop/procedure, the “n+1” values will be stored in the unew, wnew, and rho_new arrays.

Create another procedure that does the Asselin time filtering AND returns a variable in the correct arrays such that the next time step is set up.

$$\phi^{n,*} = \phi^n + 0.1(\phi^{n-1} - 2\phi^n + \phi^{n+1})$$

$$\phi^{n-1} = \phi^{n,*}$$

$$\phi^n = \phi^{n+1}$$

The second and third equations are used to do the “time flipping.”

Run the model using a “large” time step of 2 seconds and set $N=4$, i.e., there will be 8 small time steps per $2\Delta t$. Run the model for 300 seconds, plotting all 2D variables out every 100 seconds.

METG 612 2D Cloud Modeling Project

Assignment #4 – Due 31 July 2000

Today we will complete the equations for the dry portion of the model. Hand in your results to me as well as the code. (Turn in your code in electronic form, either via disk or email.) By the end of this assignment, you will now have equations of motion which look like (in differential form):

$$\frac{\partial u}{\partial t} = -u \frac{\partial u}{\partial x} - w \frac{\partial u}{\partial z} - \frac{1}{\bar{\rho}} \frac{\partial p}{\partial x} + K \nabla^2 (u - \bar{u})$$

$$\frac{\partial w}{\partial t} = -u \frac{\partial w}{\partial x} - w \frac{\partial w}{\partial z} - \frac{1}{\bar{\rho}} \frac{\partial p}{\partial z} + g \frac{\theta - \bar{\theta}}{\bar{\theta}} + K \nabla^2 w$$

$$\frac{\partial p}{\partial t} = -c_s^2 \left(\bar{\rho} \frac{\partial u}{\partial x} + \frac{\partial (\bar{\rho} w)}{\partial z} \right)$$

$$\frac{\partial \theta}{\partial t} = -u \frac{\partial \theta}{\partial x} - w \frac{\partial \theta}{\partial z} + K \nabla^2 (\theta - \bar{\theta})$$

Note that the diffusion will be done on u' , θ' rather than total u and θ . When we use soundings that have $d\theta/dz \approx 0$ and $du/dz \approx 0$ with a fixed and constant K , this methodology will prevent us from mixing the entire base state out even if no bubble is used (think about what happens if $K=75 \text{ m}^2 \text{ sec}^{-1}$ and no bubble is initialized, you still get motions because of the fixed K through the mixing term). Do this assignment using the following steps:

Add mixing to the theta equation.

Add advection to the theta equation. See if the bubble moves downward!

Add mixing to the u and w equations.

Add advection to the u and w equations.

More details:

Add mixing to the theta equation. Use the following finite difference form (2nd order):

$$ft = K(\delta_{xx}\theta + \delta_{zz}(\theta - \bar{\theta}))$$

$$ft(i, k) = K \left(\frac{\theta_{i+1,k}^{n-1} - 2\theta_{i,k}^{n-1} + \theta_{i-1,k}^{n-1}}{\Delta x^2} + \frac{\theta_{i,k+1}^{n-1,*} - 2\theta_{i,k}^{n-1,*} + \theta_{i,k-1}^{n-1,*}}{\Delta z^2} \right) \text{ where } \theta_{i,k}^{n-1,*} = \theta_{i,k}^{n-1} - \bar{\theta}_k$$

Remember that here (and for all mixing terms) we have to “backlag” the terms to the ‘n-1’ or “old” time level because using a Leap-Frog time scheme on the diffusion terms is absolutely unstable. When one backlags the diffusion terms, we are then using (by default) a forward time scheme for the diffusion terms stepping forward with a time step of $2\Delta t$ while maintaining the Leap Frog scheme for the advection terms (step 2).

Boundary conditions: Horizontal diffusion at the left and right boundary points is turned off, and vertical diffusion at the top and bottom boundaries is turned off. Note that horizontal diffusion is still done at the top and bottom boundary points, and vertical diffusion is still done at the left and right boundary points. Therefore loop indices for the x and z diffusion terms are different. Now add the time step equations for the theta equation:

$$\theta^{n+1} = \theta^{n-1} + 2\Delta t * ft$$

Run the model and make sure that your swapping of arrays for theta is correct. Add in the Asselin filter after you create the new time level of theta. Inside the bubble, you

should see it “diffusing out” and outside and well away from the bubble THERE
SHOULD BE NO CHANGE IN THETA.

Add advection to the equation. Use the following finite difference form (2nd order):

$$ft = ft_{mixing} - \overline{u\delta_x\theta^x} - \overline{w\delta_x\theta^z}$$

$$ft(i, k) = ft(i, k) - \frac{1}{2} \left(u_{i+1, k} \frac{\theta_{i+1, k} - \theta_{i, k}}{\Delta x} + u_{i, k} \frac{\theta_{i, k} - \theta_{i-1, k}}{\Delta x} \right) - \frac{1}{2} \left(w_{i, k+1} \frac{\theta_{i, k+1} - \theta_{i, k}}{\Delta z} + w_{i, k} \frac{\theta_{i, k} - \theta_{i, k-1}}{\Delta z} \right)$$

All the variables here are at the “n” time level.

Boundary conditions: At the top and bottom grid edges, set dθ/dz = 0 (W is zero at these edges anyway). That means that you will only be computing the “top” (“bottom”) half of the vertical difference term. Do the same at the right and left grid edges, setting dθ/dx = 0 (U is zero at these edges anyway).

Run the model: You should see the bubble advect downward, hit the ground, and then an outflow should move away from the center of the domain. At all times, the flow should be symmetric about the center of the domain.

Add in mixing for the u and w equations. Use the following finite difference forms (2nd order):

$$fu = K(\delta_{xx}u + \delta_{zz}(u - \bar{u}))$$

$$fu(i, k) = K \left(\frac{u_{i+1, k}^{n-1} - 2u_{i, k}^{n-1} + u_{i-1, k}^{n-1}}{\Delta x^2} + \frac{u_{i, k+1}^{n-1, *} - 2u_{i, k}^{n-1, *} + u_{i, k-1}^{n-1, *}}{\Delta z^2} \right) \text{ where } u_{i, k}^{n-1, *} = u_{i, k}^{n-1} - \bar{u}_k$$

$$fw = K(\delta_{xx}w + \delta_{zz}(w - \bar{w}))$$

$$fw(i, k) = K \left(\frac{w_{i+1, k}^{n-1} - 2w_{i, k}^{n-1} + w_{i-1, k}^{n-1}}{\Delta x^2} + \frac{w_{i, k+1}^{n-1} - 2w_{i, k}^{n-1} + w_{i, k-1}^{n-1}}{\Delta z^2} \right)$$

Again, the normal mixing at the boundaries = 0, just as in the theta equation. The effect of adding in these terms will be to reduce the magnitudes of the minimum and maximum values of u and w. Make sure that you continue to add in the buoyancy term.

Add in advection for the u and w equations. Use the following finite difference forms (2nd order):

$$\begin{aligned}
 fu &= fu_{\text{mixing}} - \overline{\overline{u}}^x \delta_x u - \overline{\overline{w}}^x \delta_z u \\
 fu(i, k) &= fu(i, k) - \frac{1}{2} \left(\left(\frac{u_{i+1, k} + u_{i, k}}{2} \right) \frac{u_{i+1, k} - u_{i, k}}{\Delta x} + \left(\frac{u_{i, k} + u_{i-1, k}}{2} \right) \frac{u_{i, k} - u_{i-1, k}}{\Delta x} \right) \\
 &\quad - \frac{1}{2} \left(\left(\frac{w_{i, k+1} + w_{i-1, k+1}}{2} \right) \frac{u_{i, k+1} - u_{i, k}}{\Delta z} + \left(\frac{w_{i, k} + w_{i-1, k}}{2} \right) \frac{u_{i, k} - u_{i, k-1}}{\Delta z} \right) \\
 fw &= fw_{\text{buoyancy}} + fw_{\text{mixing}} - \overline{\overline{u}}^z \delta_x w - \overline{\overline{w}}^z \delta_z w \\
 fw(i, k) &= fw(i, k) - \frac{1}{2} \left(\left(\frac{u_{i+1, k} + u_{i, k}}{2} \right) \frac{w_{i+1, k} - w_{i, k}}{\Delta x} + \left(\frac{u_{i, k} + u_{i-1, k}}{2} \right) \frac{w_{i, k} - w_{i-1, k}}{\Delta x} \right) \\
 &\quad - \frac{1}{2} \left(\left(\frac{w_{i, k+1} + w_{i, k}}{2} \right) \frac{w_{i, k+1} - w_{i, k}}{\Delta z} + \left(\frac{w_{i, k} + w_{i, k-1}}{2} \right) \frac{w_{i, k} - w_{i, k-1}}{\Delta z} \right)
 \end{aligned}$$

All the variables used for advection are from the “n” time level. Doing the advection on a staggered grid requires very careful attention to where you are doing the various averaging and differencing, so make sure you understand what is going on here before you plow into it. Remember, the forward differencing algorithm (FDA) is always “centered” on the variable you are computing the tendency for.

Boundary conditions: These are exactly the same as in the theta equations. At the top and bottom grid edges, set $du/dz = 0$ (W is zero at these edges anyway). That means that you will only compute the “top” (“bottom”) half of the vertical difference term. Do the same at the left and right grid edges, setting $dw/dx = 0$ (U is zero at these edges anyway). You do NOT compute tendencies for fu at the left or right grid edge, or for fw at the top and bottom grid edges.

Got it all working? Congratulations! You have just built your first Navier-Stokes model! Run the code from $t = 0$ to $t = 900$ seconds, plotting the solutions at 0, 300, 600, and 900 seconds. Make sure you use a fine enough contour interval.

Next week: Moisture and making a cloud!

METG 612 2 D Cloud Modeling Project

Assignment #5 – Due 7 August 2000

This week we complete the cloud modeling assignment by adding in equations for moisture. For this assignment, we will add in two equations for moisture, one for water vapor and one for cloud water (no rainfall).

$$\frac{\partial q_v}{\partial t} = -u \frac{\partial q_v}{\partial x} - w \frac{\partial q_v}{\partial z} + K \nabla^2 (q_v - \bar{q}_v) + M_{q_v}$$

$$\frac{\partial q_c}{\partial t} = -u \frac{\partial q_c}{\partial x} - w \frac{\partial q_c}{\partial z} + K \nabla^2 (q_c) + M_{q_c}$$

$$\frac{\partial \theta}{\partial t} = -u \frac{\partial \theta}{\partial x} - w \frac{\partial \theta}{\partial z} + K \nabla^2 (\theta - \bar{\theta}) + M_{\theta}$$

Where the “M” terms represent the microphysical processes. You will need to change the domain size and sounding for this simulation, as we need a bigger and deeper domain as well as a conditionally unstable sounding. Make the domain 40 km wide and 15 km deep, and use 1 km horizontal resolution with 500 m vertical resolution. You can increase the time step to 4 seconds, and keep NS = 4. Use the second sounding you programmed as the input sounding. Now we need to change the initial bubble specifications to have the following form and parameters:

$$\theta(x, z) = \bar{\theta}(z) + \Delta\theta * 0.5 * (\cos(\pi * r) + 1)$$

$$\text{where } r = \min \left[\sqrt{\left(\frac{x - x_o}{x_r} \right)^2 + \left(\frac{z - z_o}{z_r} \right)^2}, 1 \right]$$

$$\Delta\theta = 5 \text{ K}, x_o = 20000 \text{ m}, z_o = 1500 \text{ m}, x_r = 5000 \text{ m}, z_r = 1500 \text{ m}$$

This will specify a warm elliptical bubble in the center of the domain with a horizontal axis 10 km wide and a vertical axis 3 km deep.

Add in the two equations above using the same routines as for the theta equations, i.e., use identical finite difference methods (advection, mixing, time filtering) for q_v and q_c . Then, after you have completed all the dynamical integrations of the model, you will add in a microphysics routine (call it micro, or Kessler, or something), which will do the simple condensation and the evaporation of cloud water. Important: if, because of the numerical dispersion errors in the Leap-Frog advection, the values of q_v or q_c become less than zero, there will be a problem in the microphysics. Therefore, add in code to guarantee the *positive definiteness* of q_v and q_c .

Microphysics routine

For all points in the domain, use Teton's formula to compute the saturation mixing ratio:

$$q_{vs} = \frac{380}{\bar{p}(z)} \exp\left(17.27 \frac{(\bar{\pi}\theta^{n+1} - 273)}{(\bar{\pi}\theta^{n+1} - 36)}\right) \text{ where } \theta \text{ is the total value, not the perturbation.}$$

Next, compute at each grid point the following:

$$prod = (q_v - q_{vs}) \left[1 + \frac{L * 4093 * q_{vs}}{c_p (\bar{\pi}\theta^{n+1} - 36)^2} \right]$$

This term will either compute the production of water via condensation OR, if one is subsaturated, then it will determine the maximum amount of cloud water (if any is present) to be evaporated into the air to maintain saturation.

Evaporate, if the grid point is subsaturated and $q_c > 0$, instantaneously all the cloud water the air can hold (found in step 2). If $Prod > 0$, then we are supersaturated and retain that value. If $Prod < 0$, then we are subsaturated. Evaporate the minimum of q_c and $Prod$.

$prod = \max(prod, -qc)$

Now update theta and qc and qv.

$$\theta^{n+1*} = \theta^{n+1} + \frac{L * prod}{c_p \bar{\pi}}$$

$$q_v^{n+1*} = \max(q_v^{n+1} - prod, 0)$$

$$q_c^{n+1*} = \max(q_c^{n+1} + prod, 0)$$

Run the model for 1800 seconds, plotting ALL variables (u, w, p, theta, qv, and qc) every 600 seconds

References

- Anderson, J. K., K. Droegemier, and R. B. Wilhelmson. *Simulation of the Thunderstorm Subcloud Environment*. Preprints, 14th Conf. on Severe Local Storms, Indianapolis, IN, American Meteorological Society, 147–150, 1985.
- Bridgman, Charles J. *Introduction to the Physics of Nuclear Weapons Effects*. Ft. Belvoir, VA, Defense Threat Reduction Agency, 2001
- Defense Threat Reduction Agency (DTRA). *Handbook of Nuclear Weapons Effects, Electronic Version 3*, 2002.
- Durrant, Dale R. *Numerical Methods for Wave Equations in Geophysical Fluid Dynamics*. New York: Springer-Verlag New York, Inc., 1999.
- Holton, James R. *An Introduction to Dynamic Meteorology, Third Edition*. San Diego: Academic Press, 1992.
- Houze, Robert A., Jr. *Cloud Dynamics*, San Diego: Academic Press, 1993
- Huffines, Gary. METG 612: Air Force Institute of Technology. Wright-Patterson AFB, 2000.
- Jodoin, Vincent J. *Nuclear Cloud Rise and Growth*. Air Force Institute of Technology, Wright-Patterson AFB, OH, June 1994. (AD A280 688)
- Norment, Hillyer G. *Validation and Refinement of the DELFIC Cloud Rise module*. Atmospheric Science Associates, Bedford, MA, 15 Jan 1977. (ADA 047 372)
- Norment, Hillyer G. DELFIC: *Department of Defense Fallout Prediction System. Volume I – Fundamentals*. Atmospheric Science Associates, Bedford, MA, 31 December 1979. (ADA 088 367)
- Rogers, R. R. and M. K. Yau. *A Short Course in Cloud Physics, third edition*. Burlington, MA: Butterworth-Heinemann, 1989.
- Stull, Roland B. *An Introduction to Boundary Layer Meteorology*. AH Dordrecht, The Netherlands: Kluwer Academic Publishers Group, 1988.
- Weisman, M. L., and J. B. Klemp, 1982: *The Dependence of Numerically Simulated Convective Storms on Vertical Wind Shear and Buoyancy*. Mon. Wea. Rev., 110, 504-520.

VITA

Captain Karson Sandman was born in Lewistown, Montana. He graduated from Winnett High School in Montana. He enlisted in the Air Force serving as an electronic warfare systems repair technician. He entered the University of Florida and graduated with a Bachelor of Science degree in Mechanical engineering. After working as a design engineer for one year, he obtained his commission to again serve the Air Force.

Capt Sandman was assigned to Dover AFB, Delaware where he served as a maintenance engineering officer and a construction element chief. He next attended Explosive Ordnance Disposal (EOD) School at Eglin AFB, Florida and Indian Head NAS at Indian Head, Maryland. He was subsequently assigned as Ramstein AB EOD Flight Chief. During that time, he deployed for Operations ALLIED FORCE and ENDURING FREEDOM to Kosovo and Qatar respectively. He was then assigned as Readiness Flight Chief at Cannon AFB, NM. From there he deployed in support of Operation IRAQI FREEDOM to Kuwait and was then assigned as Civil Engineering Squadron Commander at Talil AB, Iraq.

He entered the Air Force Institute of Technology, Graduate School of Engineering and Management, and upon graduation, will be assigned to the Defense Threat Reduction Agency in Arlington, Virginia.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to the Department of Defense, Executive Services and Communications Directorate (0704-0188). Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.</p>					
1. REPORT DATE (DD-MM-YYYY) 21-03-2005		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From - To) Jun 2004 - Mar 2005	
4. TITLE AND SUBTITLE DEVELOPMENT OF 2-DIMENSIONAL CLOUD RISE MODEL TO ANALYZE INITIAL NUCLEAR CLOUD RISE			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Sandman, Karson A., Captain, USAF			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, WPAFB, OH 45433-7765			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/ENP/GNE/05-11		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Atmospheric Sciences Division Air Force Technical Applications Center Attn: Lt Col Vincent Jodoin DSN: 854-5352 Patrick AFB, FL 32925-3002 email: vincent.jodoin@patrick.af.mil			10. SPONSOR/MONITOR'S ACRONYM(S) AFTAC		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The objective of this research is to create a two-dimensional cloud rise model that could be used instead of the current 1-D cloud rise model in the Defense Land Fallout Interpretive Code (DELFI) option of the Hazard Prediction and Assessment Capability (HPAC). The model includes numerical analysis of partial differential equations involving pressure, potential temperature, horizontal and vertical winds, and specific humidity. The 2-D model developed provides a much more detailed definition of the physical properties within the mushroom cloud than the 1-D DELFI option. This is particularly useful in fallout studies on particle formation, fractionation, and particle location within the rising/risen cloud. The analysis model created for this study is the result of modifications to a convective cloud simulation. The primary modification to the convective cloud model is the incorporation of initial conditions for a nuclear cloud similar to those used in DELFI's initial conditions module.					
15. SUBJECT TERMS cloud rise model, DELFI, 2-D, convective, two-dimensional, numerical analysis					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 109	19a. NAME OF RESPONSIBLE PERSON Lt Col Steven Fiorino (ENP)
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) (937) 255-3636, ext 4506

Reset

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18